
표준API 설치가이드

- 2012. 03. 07 -

| | |
|---------------------------|----|
| 1. 표준API | 1 |
| 2. 웹보안API | 6 |
| 3. 에러코드 | 9 |
| [붙임 1] 표준API 샘플프로그램 | 10 |
| [붙임 2] 신규표준API 교체방법 | 12 |



행정 전자서명 인증관리센터

〈목차〉

| | |
|---|----------|
| 1. 표준API | 1 |
| 1.1. 표준API 개요 | 1 |
| 1.2. 표준API 적용대상 | 1 |
| 1.3. 용어설명 | 1 |
| 1.4. 표준API 종류 | 1 |
| 1.5. 표준API 디렉토리 구조 | 2 |
| 1.6. 표준API 기능 | 2 |
| 1.7. 표준API 설치 | 3 |
| 1.7.1. 표준API 설치 파일 시스템에 올리기 | 3 |
| 1.7.2. 표준API 설치 위치에 옮기기 | 3 |
| 1.7.3. 표준API 라이브러리 환경설정 | 3 |
| 1.7.4. 표준API 구동 테스트 | 5 |
| 2. 웹보안API | 6 |
| 2.1. 웹보안API 디렉토리 구조 | 6 |
| 2.2. 웹보안API 설치 | 6 |
| 2.2.1. 라이브러리 등록 | 6 |
| 2.2.2. WAS 환경설정 | 6 |
| 2.2.3. 웹보안API 환경설정 파일 | 7 |
| 2.3. 웹보안API Demo 구동 테스트 | 8 |
| 3. 예리코드 | 9 |
| [붙임 1] 표준API Sample 프로그램 | 10 |
| [붙임 2] 신규 표준API 교체 방법 [버전 1.3.0.0 => 1.4.0.0] | 12 |

1 표준API

1.1. 표준API 개요

인터넷상에서 공무원 및 행정기관 신원확인, 전자문서 위변조 방지 등을 보장하고 전자문서의 안정적 유통을 위하여 개발된 API입니다.

1.2. 표준API 적용대상

각급 행정기관

1.3. 용어설명

- 암호알고리즘 : 정보통신망에서 소통되는 중요 전자문서의 비밀성 확보를 위해 원문 암호화 및 복호화에 사용되는 방법 및 절차.
- 전자서명 : 전자문서의 작성기관 및 변경여부를 확인 할 수 있도록 전자서명 알고리즘을 이용하여 개인키로 생성한 정보로서 당해 전자문서에 고유한 것.
- 개인키 : 전자서명을 생성하기 위하여 이용하는 소유자만 사용할 수 있는 전자적 정보.
- 공개키 : 개인키에 의해 생성된 전자서명을 검증하기 위하여 이용하는 전자적 정보.
- 인증서 : 공개키가 기관 또는 담당자가 소유하는 개인키에 합치된다는 사실 등에 대하여 행정인증기관이 확인·증명하는 전자적 정보

1.4. 표준API 종류

| 배포모듈명 | 배포버전 | 개발언어 | 운영체제 / WAS | | |
|--------------------------------------|--------|----------------|---------------------------------|---|--------------------------|
| 표준API (gpkiapiV1.5.1) | V1.5.1 | C++/JAVA | Windows | 98 2000 2003 XP Vista Windows7 | 32bits/ 64bits |
| | | | | | |
| | | | HP-UX | 11.0.0 11i 11.11 11.23 ia64 | 32bits/ 64bits |
| | | | | | |
| | | | IBM-AIX | AiX 4.3 | 32bits |
| | | | | AIX 5.1 AIX 5.2 AIX 5.3 | 32bits/ 64bits |
| | | | SUN OS | Solaris 5.8 Solaris 5.9 Solaris 5.10 | 32bit/ 64bits |
| | | | | Solaris 5.10 (intel계열cpu) | 32bits |
| | | | Linux | kernel 2.4.x kernel 2.6.x ia64 | 32bits/ 64bits 개발중 |
| | | | | | |
| 웹보안API (gpkisecureweb.v2.0.0.9) | V1.5 | ASP JSP/PHP | Windows IIS, Tomcat, JEUS 등 | | |
| | | | Weblogic 8.X 9.X, JEUS 4.X 5.X | | |
| | | | Websphere, Tomcat 4.X 5.X 6.X 등 | | |

1.5. 표준API 디렉토리 구조

| 구분(폴더) | | 파일명 | 설명 |
|--------------------|-------------|-----------------------------|------------------------------------|
| 표준 API (v1.5.1) | conf | gpkiapi.conf | 인증서 검증에 필요한 정보 포함 환경파일 |
| | | gpkiapi.lic | 표준API 라이선스 |
| | info | gpkiapi_info | 표준 API 버전을 확인할수 있는 프로그램 |
| | jar | libgpkiapi_jni.jar | 표준API jar 파일 |
| | javadoc | | 표준API 설치적용가이드(html) |
| | jtest | /java | 테스트 Sample 자바코드 |
| | | /class | 테스트 Sample 실행파일 |
| | lib / lib64 | gpkiapi.dll | 윈도우용 |
| | | libgpkiapi.so | Sun, Linux, UnixWare |
| | | libgpkiapi.a | IBM AIX |
| | | libgpkiapi.sl | HP UX |
| | | 기타 | ldap 라이브러리등 표준 API에서 참조하는 라이브러리 파일 |
| | Sample | | jtest 또는 ctest시 테스트 파일 |

1.6. 표준API 기능

| 기능 | 설명 |
|--------------------------|---|
| 인증서 정보 확인 모듈 | X.509 인증서의 주요 필드 정보 확인 |
| 개인키 모듈 | 인증서 소유자의 중요한 정보인 개인키의 암호화/복호화 |
| 저장매체 모듈 | 인증서 개인키를 저장매체로부터 읽기, 저장, 삭제 |
| 유선용 전자서명, 암호메세지 생성/처리 모듈 | 유선환경에서 보안서비스를 제공하기 위해서 사용하는 보안 메시지 생성/처리 |
| 무선용 전자서명, 암호메세지 생성/처리 모듈 | 무선환경에서 보안서비스를 제공하기 위해서 사용하는 보안 메시지 생성/처리 |
| 시점확인 서비스 이용 모듈 | 시점확인 서버를 이용하여 특정 메시지에 대한 시점확인 토큰 발급 받음 |
| 인증서 내 정보를 이용한 본인확인 모듈 | 인증서 내에 포함되어 있는 정보를 이용하여 인증서 소유자에 대한 본인 확인 |
| 보안 알고리즘 모듈 | 다양한 보안서비스 제공을 가능하게 하기 위한 보안 알고리즘 |
| BASE64 모듈 | BASE64 인코딩 / 디코딩 |
| 디렉토리 접근 모듈 | 인증서와 인증서 폐지목록과 같은 디렉토리 서버에 게시되어있는 데이터 획득 |
| 객체인증 모듈 | 상대방을 확인하기 위한 객체 인증 프로토콜 |

1.7. 표준API 설치

1.7.1. 표준API 설치 파일 시스템에 올리기

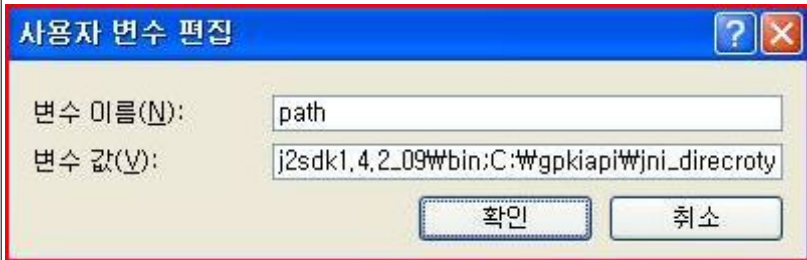
- 표준API 설치 파일을 보안 서비스를 제공할 시스템에 복사합니다. FTP 로 설치파일을 올릴 때는 반드시 "binary" 모드로 올려야 합니다.

1.7.2. 표준API 설치 위치에 옮기기

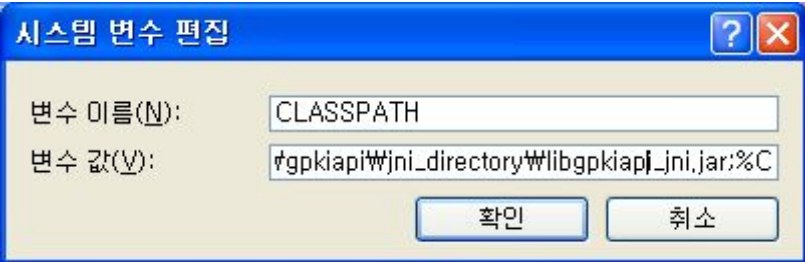
- 표준API 가 구동될 위치를 정하고 표준API 모듈 (ex. libgpkiapi.so, libgpkiapi_jni.jar) 과 환경파일 (gpkiapi.conf)을 해당 위치로 옮깁니다. (만약, 표준API에 포함되어 있는 LDAP 라이브러리가 해당 시스템에 설치되어 있지 않다면 LDAP 라이브러리도 함께 설치합니다.)

1.7.3. 표준API 라이브러리 환경설정

- 라이브러리 경로 설정

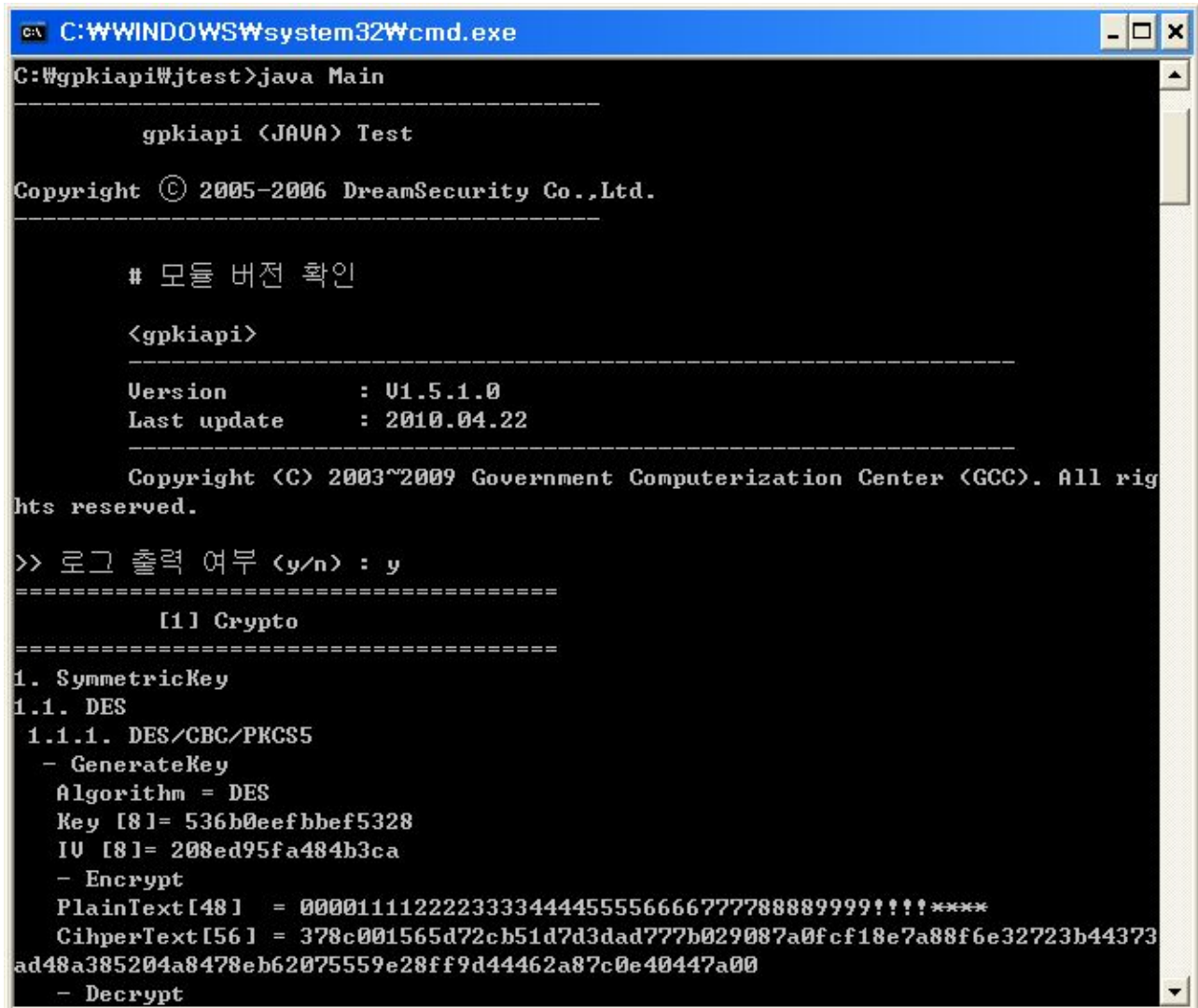
| | | |
|-------------------------|---|---|
| Windows 인 경우 | <p>C/C++ 용 표준API 와 LDAP 라이브러리가 위치해 있는 경로를 환경변수에 등록합니다.</p> <p>"내 컴퓨터=>속성=>고급=>환경변수"에서 기존 "path" 변수에 "라이브러리가 설치된 디렉토리"를 설정합니다.</p>  | |
| Solaris, Unix, Unixware | C 또는 TC shell | setenv LD_LIBRARY_PATH "라이브러리가 설치된 디렉토리" |
| | Corn shell | export LD_LIBRARY_PATH="라이브러리가 설치된 디렉토리" |
| | Born shell | LD_LIBRARY_PATH="라이브러리가 설치된 디렉토리" export LIBPATH |
| IBM AIX | C 또는 TC shell | setenv LIBPATH "라이브러리가 설치된 디렉토리" |
| | Corn shell | export LIBPATH="라이브러리가 설치된 디렉토리" |
| | Born shell | LIBPATH="라이브러리가 설치된 디렉토리" export LIBPATH |
| HP UX | C 또는 TC shell | setenv SHLIB_PATH "라이브러리가 설치된 디렉토리" |
| | Corn shell | export SHLIB_PATH="라이브러리가 설치된 디렉토리" |
| | Born shell | SHLIB_PATH="라이브러리가 설치된 디렉토리" exprot SHLIB_PATH |

- classpath 설정(자바 사용할 경우)

| | | |
|-------------------------|---|--|
| Windows 인 경우 | <p>java용 표준API의 libgpkapi_jni.jar가 위치해 있는 경로를 환경변수에 등록합니다.</p> <p>“내 컴퓨터=>속성=>고급=>환경변수”에서 기존 “CLASSPATH” 변수에 “라이브러리가 설치된 디렉토리/libgpkapi_jni.jar.”를 설정합니다.</p>  | |
| Solaris, Unix, Unixware | C 또는 TC shell | setenv CLASSPATH "라이브러리가 설치된 디렉토리/libgpkapi_jni.jar." |
| | Corn shell | export CLASSPATH="라이브러리가 설치된 디렉토리/libgpkapi_jni.jar." |
| | Born shell | CLASSPATH="라이브러리가 설치된 디렉토리/libgpkapi_jni.jar." export CLASSPATH |
| IBM AIX | C 또는 TC shell | setenv CLASSPATH "라이브러리가 설치된 디렉토리/libgpkapi_jni.jar." |
| | Corn shell | export CLASSPATH="라이브러리가 설치된 디렉토리/libgpkapi_jni.jar." |
| | Born shell | CLASSPATH="라이브러리가 설치된 디렉토리/libgpkapi_jni.jar." export CLASSPATH |
| HP UX | C 또는 TC shell | setenv CLASSPATH "라이브러리가 설치된 디렉토리/libgpkapi_jni.jar." |
| | Corn shell | export CLASSPATH="라이브러리가 설치된 디렉토리/libgpkapi_jni.jar." |
| | Born shell | CLASSPATH="라이브러리가 설치된 디렉토리/libgpkapi_jni.jar." exprot CLASSPATH |

1.7.4. 표준API 구동 테스트

- command 창 실행 => sample 프로그램 이동 (java) => java Main 실행



```
C:\WINDOWS\system32\cmd.exe
C:\Wgpkapi\Wjtest>java Main

-----
      gpkapi <JAVA> Test
Copyright © 2005-2006 DreamSecurity Co.,Ltd.
-----

# 모듈 버전 확인

<gpkapi>
-----
Version       : 01.5.1.0
Last update   : 2010.04.22
-----

Copyright (C) 2003~2009 Government Computerization Center (GCC). All rights reserved.

>> 로그 출력 여부 (y/n) : y
=====
      [1] Crypto
=====
1. SymmetricKey
1.1. DES
1.1.1. DES/CBC/PKCS5
- GenerateKey
  Algorithm = DES
  Key [8] = 536b0eefbbef5328
  IV [8] = 208ed95fa484b3ca
- Encrypt
  PlainText[48] = 0000111122223333444455556666777788889999!!!!****
  CipherText[56] = 378c001565d72cb51d7d3dad777b029087a0fcf18e7a88f6e32723b44373ad48a385204a8478eb62075559e28ff9d44462a87c0e40447a00
- Decrypt
```

[sample 프로그램 실행]

2 웹 보안API

2.1. 웹 보안API 디렉토리 구조

| 구분(폴더) | 설명 |
|------------------------------|---------------|
| gpkisecureweb (v.2.0.0.9) | certs |
| | conf |
| | demo |
| | doc |
| | gpkisecureweb |
| | lib |
| | log |

2.2. 웹 보안API 설치

2.2.1. 라이브러리 등록

- gpkisecureweb.jar, libgpkapi_jni.jar(표준API) 라이브러리 등록

| WAS 구분 | 설치 위치 |
|----------|-----------------------------|
| Tomcat | \$TOMCAT_HOME/shared/lib |
| JEUS | \$JEUS_HOME/lib/application |
| WebLogic | \$WEBLOGIC_HOME/lib |

2.2.2. WAS 환경설정

1) Tomcat

- \$Tomcat_HOME/bin/catalina.sh 또는 Catalina.bat 에 아래 부분을 추가합니다.
set JAVA_OPTS="-Dcom.dsddf.jdf.config.file=/gpkisecureweb/conf/dsddf.properties"

2) JEUS

- \$Jeus_HOME/config/\$hostname/JEUSMain.xml 파일에 아래 부분을 추가합니다.
<command-option>
-Dcom.dsddf.jdf.config.file=/gpkisecureweb/conf/dsddf.properties
</command-option>

3) WebLogic

- \$WebLogic_HOME/workshop/workshop.sh 또는 workshop.cfg 파일에 아래 부분을 추가합니다.
/bea/java141_05/jre/bin/java "-XX:-UseThreadPrioritite -Xmx256m -Xms64m -client
-Djava.system.class.loader="workshop.core.AppClassLoader"
-Dcom.dsddf.jdf.config.file="/gpkisecureweb/conf/dsddf.properties"

[현재 많이 사용되는 WAS 에 대한 예]

2.2.3. 웹보안API 환경설정 파일

- \$gpkisecureweb/conf/dsjdf.properties
- \$gpkisecureweb/conf/gpkisecureweb.properties
- ~(Web Root)/gpkisecureweb/install.html
- ~(Web Root)/gpkisecureweb/var.js
- ~(Web Root)/gpkisecureweb/setup/setup.conf

※ \$: 웹용 표준API 설치 위치, ~(Web Root): 웹 홈디렉토리

- \$gpkisecureweb/conf/dsjdf.properties

| 항목 (변수) | 설명 |
|--|-----------------|
| logger.dir=[gpkisecureweb]/log | Log 관련 경로 지정 |
| pbf.propertiesFile=[gpkisecureweb]/conf/gpkisecureweb.properties | 프로젝트 설정파일 경로 지정 |

- \$gpkisecureweb/conf/gpkisecureweb.properties

| 항목 (변수) | 설명 |
|--|--|
| GPKISecureWeb.crypto.algo=SEED/CBC | 암호알고리즘 세팅 |
| GPKISecureWeb.errorPage=/gpkisecureweb/GPKIError.jsp | 에러페이지 설정 |
| GPKISecureWeb.CertFilePathName= GPKISecureWeb.PrivateKeyFileName= GPKISecureWeb.PrivateKeyPasswd= | GPKI 서버인증서 위치 (인증서파일, 인증서키파일, 비밀번호)서버인증서는 행정안전부를 통해 발급받아 설치 후 설정해야하며 변경시 WAS 를 재구동해야함. |
| GPKISecureWeb.gpkiaapi.ConfFilePath= | 표준API con/gpkiaapi.conf 파일이 있는 절대경로 지정 |
| GPKISecureWeb.AnyPolicy=yes GPKISecureWeb.Policy=1 2 410 20005 1 1 4 | 인증서 정책 검증 설정 |
| GPKISecureWeb.VerifyCertMethod=CRL OCSP IVS | 인증서 검증 방법 |
| GPKISecureWeb.TrustedROOTCACert.count=2 | ROOTCA 인증서의 개수 |
| GPKISecureWeb.TrustedROOTCACert.FilePathName.1=C:/gpkisecureweb/certs/NPKIRootCA1.der GPKISecureWeb.TrustedROOTCACert.FilePathName.2=C:/gpkisecureweb/certs/GPKIRootCA1.der | ROOTCA 인증서의 위치 (NPKI, GPKI) |

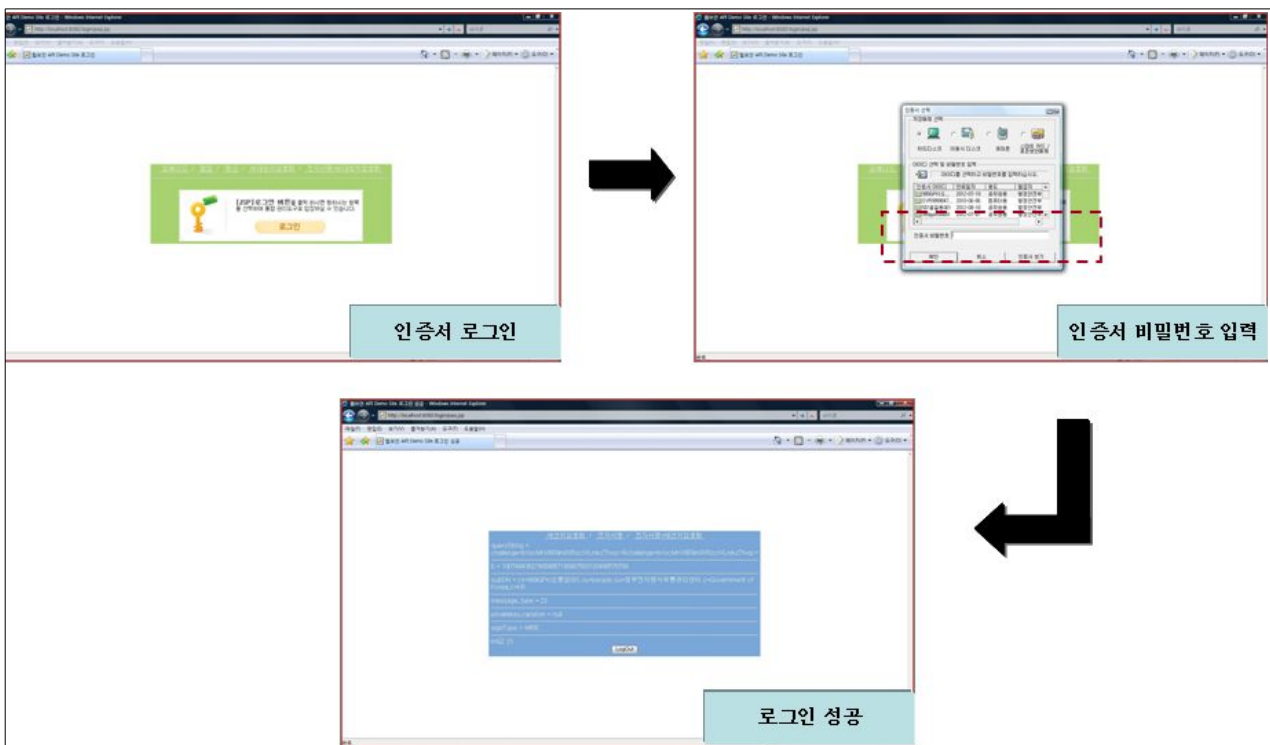
- ~(Web Root)/gpkisecureweb/install.html

```
<script language='javascript' src='/gpkisecureweb/var.js'></script>
<script language='javascript' src='/gpkisecureweb/install.js'></script>
```

- ~(Web Root)/gpkisecureweb/var.js

| | |
|--|---|
| ServerCert = "MIID5" | 서버인증서세팅: 서버용인증서(암호용)의 Base64Encode 값 세팅 => [붙임1 참조] |
| AlogMode = 0x30 | 암복호화 알고리즘 (0x20:3DES, 0x30:SEED, 0x40:NEAT, 0x50:ARIA) |
| CNCertType = 0x00 | 인증서 사용자 인터페이스에 로딩할 인증서 종류 세팅 (GPKI, NPKI:0x00, GPKI:0x01, NPKI:0x02) |
| ServerAddr = "10.1.1.1:8080" | 웹서버 IP : Port |
| CodeBase_GPKIInstaller = "CODEBASE='http://'+ServerAddr+'/gpkisecureweb/ gpkisecureweb/setup/GPKISecureWebX.cab#version=2,0,0,9'" | 실제 서비스를 하는 웹서버 IP:Port 설정 및 GPKISecureWebX.cab 파일 위치 지정 |
| var Object_GPKIInstaller = "<OBJECT ID='GPKISecureWebX' CLASSID= 'CLSID{8223F3A-1420-4245-88F2-D874FC081572}' width=0 height=0"; Object_GPKIInstaller += CodeBase_GPKIInstaller; Object_GPKIInstaller += "</OBJECT>"; | GPKIInstaller ActiveX지정 |

2.3. 웹보안API Demo 구동 테스트



* 웹보안API는 표준API가 먼저 설치되어 있어야 합니다.

3

에러코드

- Exception 발생 시, 에러 메시지에 포함되어있는 에러코드는 1000번부터 ~ 5000번 까지 나올 수 있으며, 에러코드 범위 별로 다음과 같은 의미를 가지고 있습니다.

| 에러코드 범위 | 설명 |
|-------------|---------------------------------------|
| 1000 ~ 1099 | API 초기화와 API 사용 중 포괄적으로 일어날 수 있는 에러코드 |
| 1100 ~ 1199 | 라이선스 검증과 관련한 에러코드 |
| 1200 ~ 1299 | 인증서 검증과 정보 조회와 관련한 에러코드 |
| 1300 ~ 1399 | 개인키 암호/복호와 관련한 에러코드 |
| 1400 ~ 1499 | 저장매체와 관련한 에러코드 |
| 1500 ~ 1599 | 유무선 서명, 암호 메시지와 관련한 에러코드 |
| 1600 ~ 1699 | 시점확인 서비스와 관련한 에러코드 |
| 1700 ~ 1799 | 본인확인과 관련한 에러코드 |
| 1800 ~ 1899 | 보안 알고리즘과 관련한 에러코드 |
| 1900 ~ 1999 | BASE64 인코딩/디코딩과 관련한 에러코드 |
| 2000 ~ 2099 | 디렉토리 서버(LDAP)와 관련한 에러코드 |
| 2100 ~ 2199 | 통합검증 서버와 관련한 에러코드 |
| 4000 ~ 4099 | 객체인증 프로토콜과 관련한 에러코드 |

붙임 1

표준API Sample 프로그램 코드

* 표준API 가이드 참조

- java : ~/gpkiapiV1.5.1/javadoc/index.chm
- C : ~/gpkiapiV1.5.1/doc/manual.pdf

* 웹보안API 가이드 참조 [~/gpkisecureweb/doc/웹용 표준보안 API 매뉴얼(유선) v2.0.0.8.chm]

가) 인증서 Base64Encoding

```
<%@ page import="com.gpki.gpkiapi.storage.Disk" %>
<%@ page import="com.gpki.gpkiapi.util.Base64" %>
<%@ page import="com.gpki.gpkiapi.cert.X509Certificate" %>

<%
    String SERVER_KM_CERT_PATH = "서버인증서파일경로/서버인증서파일(SVR_env.cer)";
    Base64 base64 = new Base64();
    byte[] bBase64 = null;
    String strBase64 = "";

    X509Certificate srvCert = Disk.readCert(SERVER_KM_CERT_PATH);
    bBase64 = srvCert.getCert();
    strBase64 = new String(base64.encode(bBase64));
%>
```

나) Ldap에서 인증서키를 구함

```
<%@ page import="com.gpki.gpkiapi.GpkiApi" %>
<%@ page import="com.gpki.gpkiapi.cert.X509Certificate " %>
<%@ page import="com.gpki.gpkiapi.util.Ldap" %>
<%@ page import="com.gpki.gpkiapi.storage.Disk " %>
<%@ page import="com.gpki.gpkiapi.util.Base64 " %>

<%
    GpkiApi.init(".");
    X509Certificate recCert1;

    Ldap ldap = new Ldap();
    ldap.setLdap("ldap도메인",389);
```

```
//인증키를 ldap에서 구함
ldap.searchCN(ldap.DATA_TYPE_KM_CERT,"서버인증서 ID");
byte[] sn=ldap.getData() ;

// 인증서
recCert1 = Disk.readCert(path);
%>
```

붙임 2 신규 표준API 교체 방법[버전 1.3.0.0 => 1.4.0.0/1.5.x]

- 컴퓨팅 기술의 급속한 발달로 인해 기존 인증서 암호 알고리즘의 안전성이 저하되어 보다 고도화된 암호 알고리즘이 요구됨으로 전자서명키 길이 상향 조정, 해쉬 알고리즘 교체 등 인증서 암호체계 고도화 적용에 따라 표준API 변경에 대한 내용을 기술함 (기 설치한 표준API 를 새로 배포하는 표준API 로 재설치)

1) 라이브러리 변경

* 기존라이브러리 백업 또는 삭제 후 새로 배포하는 라이브러리 복사

| 구분 | 기존 라이브러리 | 변경 라이브러리 |
|--------------------------|--|--|
| Windows | gpkapi.dll | gpkapi.dll |
| | gpkapi_jni.dll | |
| Solaris, Linux, Unixware | liblber.so | |
| | libgpkapi.so | libgpkapi.so |
| | libgpkapi_jni.so | |
| IBM AIX | libgpkapi.a | libgpkapi.a |
| | libgpkapi_jni.a | |
| HP UX | liblber.sl | liblber.sl |
| | libgpkapi.sl | libgpkapi.sl |
| | libgpkapi_jni.sl | |
| WAS | libgpkapi_jni.jar gpkisecureweb.jar | libgpkapi_jni.jar gpkisecureweb.jar |

* 라이브러리 변경 후 가이드의 1.7.4 표준API 구동테스트 필수

2) 라이선스 추가

- 인증체계 고도화에 따라 재배포 이전의 표준API 설치 운영 시스템은 신규 표준API 와 같이 배포되는 라이선스를 추가하여야 합니다.
- 디렉토리 위치 (라이선스 파일 : gpkapi.lic) 는 표준API 설치 위치의 conf 디렉토리에 복사
가) 표준API

~/libgpkapi/conf/gpkapi.lic

나) 웹보안API

~/gpkisecureweb/conf/gpkapi.lic

* 행정전자서명 적용 시스템 운영 서버1식(IP)에 라이선스 1개 적용합니다.