http://www.egovframe.go.kr/wiki/doku.php?id=egovframework:hyb3.5:guide:add:npki_%EC%BC%80%EC%9D%B4%EC%82%AC%EC%9D%B8

NPKI(WizSign) Device API Guide Program

Outline

NPKI(WizSign) is a guide application for eGov Device API, using the mobile device API framework to be used as a tool and a reference when developing hybrid applications. It supports the NPKI related functions of mobile smart devices through JavaScript-based NPKI DeviceAPI.

Also, it connects with web server applications based on eGov standard framework in order to authenticate certificates, save the result to server, and reference authentication results log.

Feature

This Guide Program provides **Select/authenticate Mobile Device Certification** and **View authentication log information** features. These features are realized in a way that applies **Standard Security API** to web server applications that allows for checking certificate information.

Assumptions

Category	Description
Local Device Environments	eGovFramework Runtime Environment 3.5, Android SDKAPI 22(version 5.0 Lollipop)
Server-side Developmental Environment	eGov standard framework developmental environment 3.5 Standard Security API setting (refer to "Server Application" section on the settings)
Works in sync with Mash up Open API	N/A
Test Device	Galaxy S2
Test Platform	Android 2.3
Libraries Added	NPKI WizSign module
Using cross domain	When using certain outside domains or its subdomains on PhoneGap, add such domains on <access origin="" /> at Res/xml/config.xml.

NPKI WizSign Library

File name	Description
libs/jldap-4.3.jar	WizSign library

libs/KSignCrypto_Applet.jar	WizSign library
libs/WizSign.jar	Core library that conducts functions related to Electronic signature, security channel, and certificates.
assets/www/js/egovframework/mbl/hyb/wizsignpg.js	JavaScript interface for using WizSign API on PhoneGap based Hybrid application

Restriction

Restrictions on NPKI technology cooperation

The code libraries below from NPKI device API component library have their own license as a security Native Module. Therefore, libraries on the table below are omitted from NPKI device API distribution, and any government entities or firms intending on testing or operating such modules must contact the firm listed below.

Name	Point of Contact	Contact Information	E-Mail
KSign Inc.	Shin, Dong-Soo	02-564-0182	men4u@ksign.com

Applying eGov security standard API

A separate request for security standard API must be made in order to use eGov security standard API, which can be made at Administrative Electronic Signature Management Center (http://www.gpki.go.kr).

Follow the instructions below.

▶ When Standard API management system can be accessed

⬭ Request the API via web at [Standard API management system] (attach memorandum and diagram)

⬭ Service URL : http://api.gpki.go.kr

Send memo to Korea Local Information Research & Development Institute - Local Information Center - Information Infrastructure Branch.

The content of the memorandum should include the name of the system, Point of Contact, and the request for standard API.

- The following service can only be accessed in the government network -

▶ When commercial internet (http://api.gpki.go.kr Connection Unavailable) cannot be used

⬭ At the Government Electronic Signature Certification Management Center (http://www.gpki.go.kr) website, fill in the request form (“Downloads-Certification Request Forms-7.Standard API request instructions and Standard API request form”) along with the memorandum.

Memorandum To : Korea Local Information Research & Development Institute - Local Information Center - Information Infrastructure Branch

The content of the memorandum should include the name of the system, Point of Contact, and the request for standard API.

Refer to Government Electronic Signature Certification Management Center(http://www.gpki.go.kr) for additional information and inquiries.

Supported devices and platforms

N/A

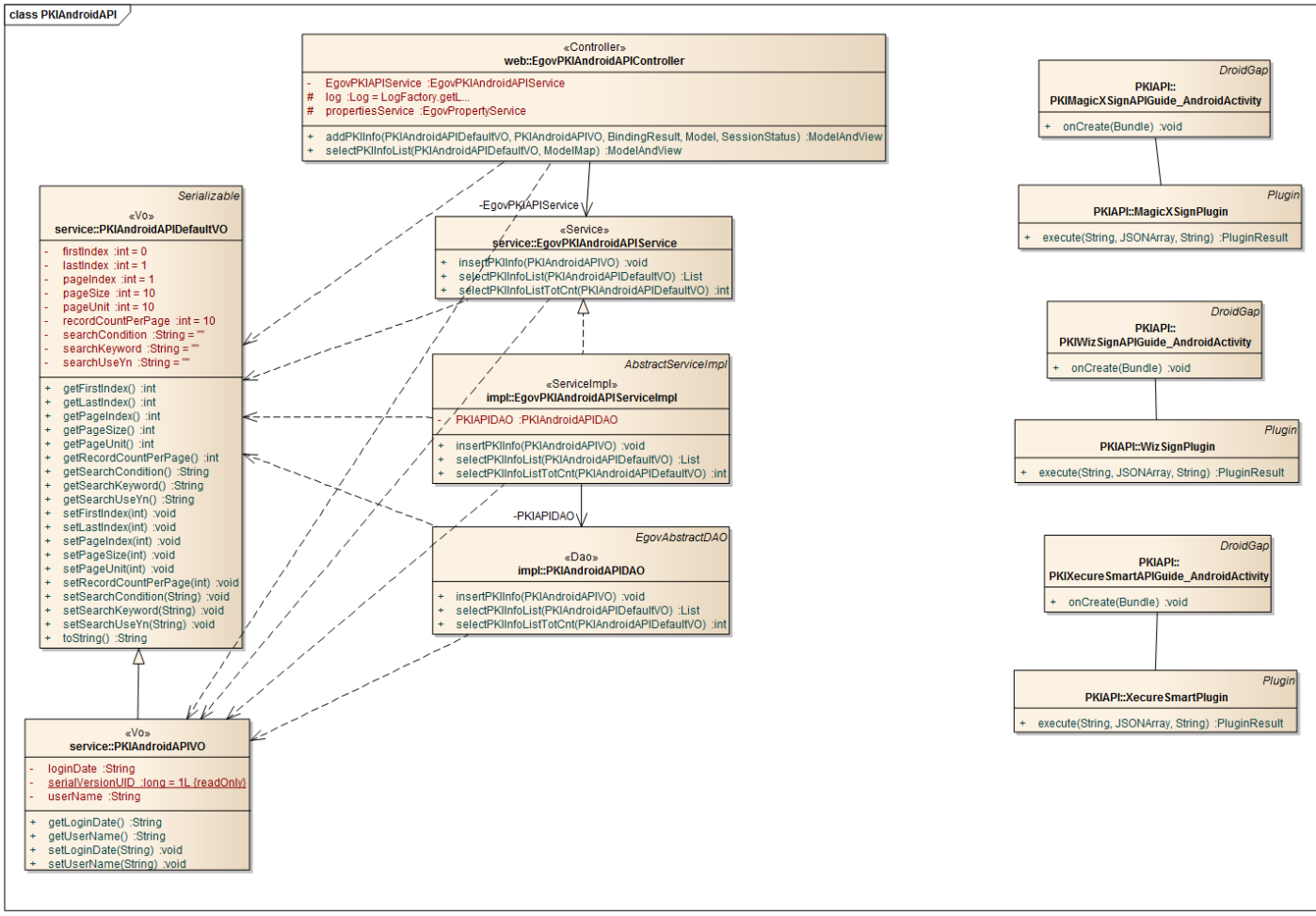
Supported devices and platforms

N/A

Description

NPKI Device API Guide Program is comprised of: a) a function that selects the certificate on the mobile device and then creates the signature data, sends it, and authenticates the certificate and b) inquires the authentication log data. (refer to the Related Features section)

Related Class Diagram



Device Application

Source

Type	Title	Remark

JSJSActivity	kr.go.egovframework.hyb.pkiapi.wizsign.PKIWizSignAPIGuide_AndroidActivity	NPKI Guide Program Activity Class
CSS	assets/www/css/egovframwork/mbl/hyb/PKIWizSignAPI.css	NPKIAPI Guide Program Core Cascading Style Sheets
IMAGE	assets/www/images/egovframwork/mbl/hyb/	NPKIAPI Guide Program main Image Folder
JS	assets/www/js/egovframwork/mbl/hyb/PKIWizSignAPI.js	NPKIAPI Guide Program main JavaScript
JS	assets/www/js/egovframwork/mbl/hyb/wizsignpg.js	NPKIAPI Guide Program main JavaScript
JS	assets/www/js/egovframwork/mbl/hyb/messages_ko.js	JavaScript for Validate Message Processing
RES	assets/www/res/	NPKIAPI Guide Program main Resource folder
XML	AndroidManiFest.xml	Configuration XML for Android
HTML	assets/www/PKIWizSignAPI.html	NPKIAPI main page
HTML	assets/www/Intro.html	NPKIAPI Intro page
HTML	assets/www/license.html	NPKIAPI license page
HTML	assets/www/overview.html	NPKIAPI feature description page

Function API

WizSign API DOC

API used on the Guide Program

doSignature

- Conducts electronic signature using selected Certificate and returns signature value

- Parameters: Certificate #, Certificate password, subject original
- Return value (hash table)

'signedData' : signed data
'errMsg' : error message

```
var args = new Array();
args[0] = selectCertNum.toString() ;
args[1] = '1';
args[2] = stringToSign;

WizSignPG.doSignature(args, function(result) {
        var signedData = result['signedData'];      // signed data
    }, function(error) {
        alert(error['errMsg']);             // error message
});
```

getCertificates

- Calls and returns the saved Certificate list

- Parameters : N/A
- Return value (hash table)

'Certificates' : Certificate list
'errMsg' : error message

```
WizSignPG.getCertificates("", function(result) {
	var certList = result['Certificates'];

	for(var i=0 ; i<certList.length ; i++) {
		certList[i]['NUM'];
		certList[i]['HOST'];
		certList[i]['ISSUED BY'];
		certList[i]['EXPIRATION DATE'];
	}

	}, function(error) {
		alert(''error['errMsg']);
});
```

Certificate information hash table

Hash table	Description
NUM	Certificate No.
Version	Certificate version
Serial No.	Certificate Serial No.
Signature algorithm	Certificate Signature algorithm
Issuer	Certificate issuer information
Date of effect	Certificate's date of effect
Expiration Date	Certificate's Expiration Date
Subject	Certificate subject data
Public key algorithm	Certificate Public key algorithm
Issuer Serial No.	Issuer Serial No.
Public Key	Public Key value
Institution key identifier	Institution key identifier
Subject identifier	Subject identifier

Policy	Policy
Subject alternative name	Subject alternative name
CRL division point	CRL division point
Institution information access	Institution information access (OCSP)
Key use	Purpose of key use
Signature	Certificate signed value

verifyCertPassword

- Verifies selected Certificate's password.

- Parameters: Certificate No., Certificate password
- Return value (hash table)

'result' : Certificate password verification result('OK' when successful)
'errMsg' : error message

```
var args = new Array();
args[0] = certNum.toString();
args[1] = certPass;

WizSignPG.verifyCertPassword(args, function(result) {
	var runResult = result['result'];

	if(runResult == 'OK') {
		alert('Correct Password.');
	}

	}, function(error) {
	alert(error['errMsg']);
});
```

Server Application

Source

Type	Title	Remark
Controller	egovframework.hyb.add.pki.web.EgovPKIAndroidAPIController.java	NPKIAPI Guide Program Controller Class
Service	egovframework.hyb.add.pki.service.EgovPKIAndroidAPIService.java	NPKIAPI Guide Program Service Class
ServiceIimpl	egovframework.hyb.add.pki.service.impl.EgovPKIAndroidAPIServiceImpl.java	NPKIAPI Guide Program ServiceImpl Class
VO	egovframework.hyb.add.pki.service.PKIAndroidAPIDefaultVO.java	NPKIAPI Guide Program VO Class
VO	egovframework.hyb.add.pki.service.PKIAndroidAPIVO.java	NPKIAPI Guide Program VO Class

VO	egovframework.hyb.add.pki.service.PKIAndroidAPIXmlVO.java	NPKIAPI Guide Program XML related VO Class
DAO	egovframework.hyb.add.pki.service.impl.PKIAndroidAPIDAO.java	NPKIAPI Guide Program Dao Class
QUERY XML	resources/egovframework/sqlmap/hyb/add/pki/EgovPKIAndroidAPIGuide_SQL_XXX.xml	NPKIAPI Guide Program QUERY XML
Idgen XML	resources/egovframework/spring/context-idgen.xml	NPKIAPI Guide Program ID generation Idgen XML

Related Tables

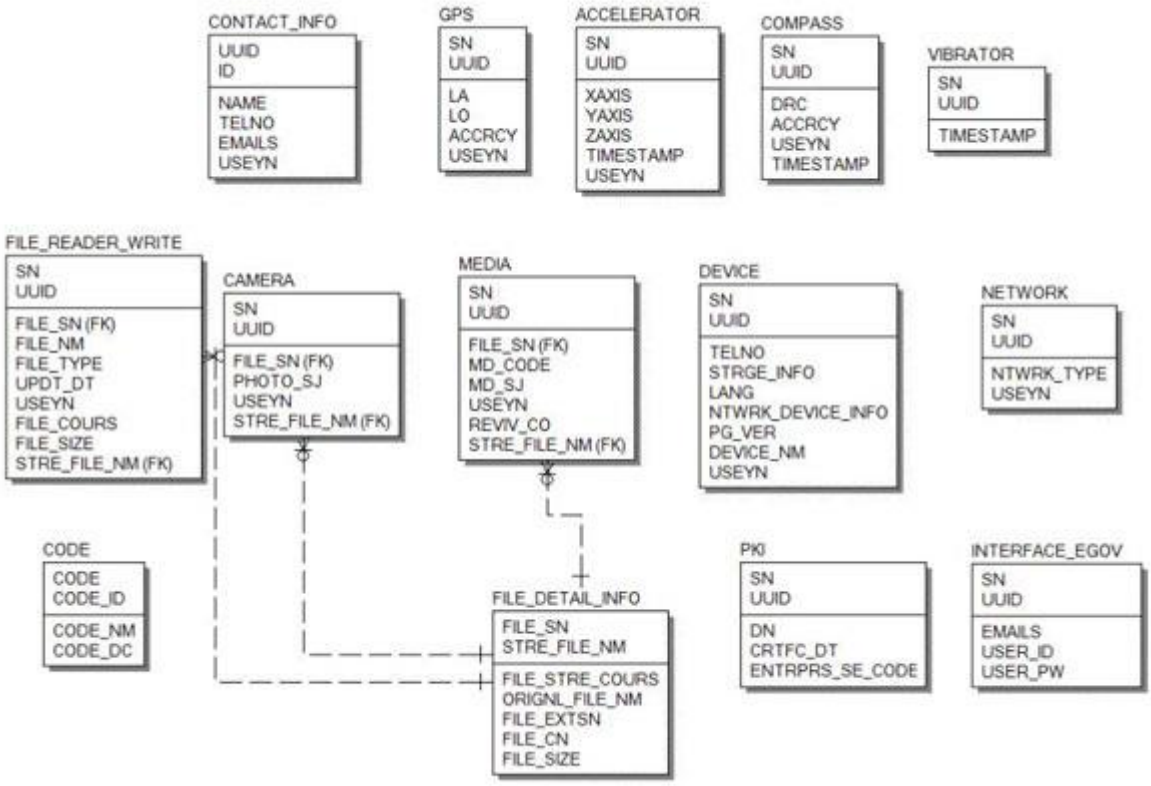
Title Table	Remark
PKI PKI	Certification Recognition Log Management

Tables Breakdown

- PKI

No.	Column	Title of Column	Type	Length	Null	KEY
1	SN	Serial No.	NUMERIC	6	NotNull	pk
2	UUID	UUID	VARCHAR	50	NotNull	pk
3	DM	DM	VARCHAR	255	Null	
4	CRTFC_DT	Date certified	DATE		Null	
5	ENTRPRS_SE_CODE	Email	DATE		Null	

ERD



Standard API for Security

```
public String verifyCert(PKIAndroidAPIVO pkiVo) throws Exception {
	// API initialization
	GpkiApi.init("C:/libgpkiapi_jni/conf");
	String sign;
	sign = pkiVo.getSign();
	return verify(Base64.decode(sign));
}

private String verify(final byte[] bSignedData)	{
	String sClientName = "";
	try {
		// authenticates signature
		SignedData signedData = null;
		signedData = new SignedData();
		signedData.verify(bSignedData);

		// acquires server's signing Certificate in order to authenticate subject's Certificate
		X509Certificate clientCert = null;
		clientCert = signedData.getSignerCert(0);

		// Certificate authentication
		CertPathValidator certPathValiditor = null;
```

```
            certPathValiditor = new CertPathValidator("C:/libgpkiapi_jni/conf/gpkiapi.conf");

            // adds top truted Certificate
            X509Certificate rootCertRsa = null;
            rootCertRsa = Disk.readCert("C:/libgpkiapi_jni/conf/root-rsa2.der");
            X509Certificate rootCertRsaSha = null;
            rootCertRsaSha = Disk.readCert("C:/libgpkiapi_jni/conf/root-rsa-sha2.der");
            certPathValiditor.addTrustedRootCert(rootCertRsa);
            certPathValiditor.addTrustedRootCert(rootCertRsaSha);

            // sets client's Certificate authentication level
            certPathValiditor.setVerifyRange(CertPathValidator.CERT_VERIFY_FULL_PATH);

            // sets verification on whether or not the client's Certificate will be purged (sets CRL/ARL
verification)
            certPathValiditor.setRevokationCheck(CertPathValidator.REVOKE_CHECK_ARL |
CertPathValidator.REVOKE_CHECK_CRL);

            // requests Certificate authentication
            certPathValiditor.validate(CertPathValidator.CERT_SIGN, clientCert);

            sClientName = clientCert.getSubjectDN();


        } catch (Exception e) {
            sClientName = "";
        }
        return sClientName;
}
```

Properties

Necessary sections and settings for using NPKI related features of mobile device, provided by NPKI Device API Guide Program, are as follows.

Device Application

res/xml/config.xml

```
<!-- PhoneGap Plugin for eGov Interface Device API Class -->
<pluginname="EgovInterfacePlugin"value="kr.go.egovframework.hyb.plugin.EgovInterfacePlugin"/
>
<!-- Phonegap Plugin class for using eGov NPKI Device API-->
<pluginname="WizSignPG"value="com.ksign.wizsign.sdk.wizsignAPI.WizSignPG"/>
```

res/values/serverinfo.xml

```
<!-- Server Directory for eGov Interface Device API Class -->
<?xmlversion="1.0"encoding="utf-8"?>
<resources>
            <stringname="SERVER_URL">http://192.168.100.222:8080/DeviceAPIGuideTotal_Web
_V1.7.1</string>
</resources>
```

AndroidManifest.xml

```
<uses-permissionandroid:name="android.permission.INTERNET"/>
<uses-permissionandroid:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permissionandroid:name="android.permission.ACCESS_NETWORK_STATE"/>
```

Server Application

pom.xml

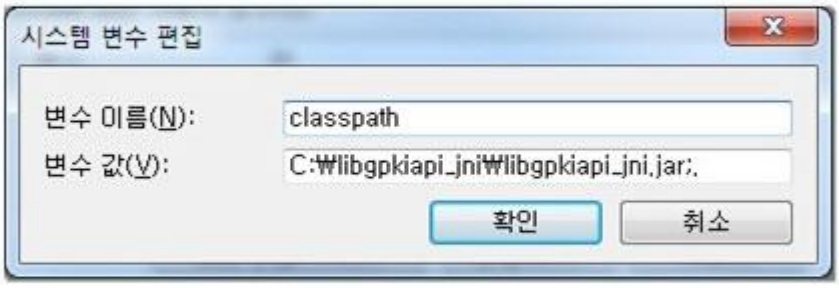
```
<dependency>
        <groupId>egovframework.com.cmm.uat</groupId>
        <artifactId>libgpkiapi_jni</artifactId>
        <version>1.4.0.0</version>
</dependency>
```

resource/egovframework/sqlmap/sql-map-config_[DB NAME].xml

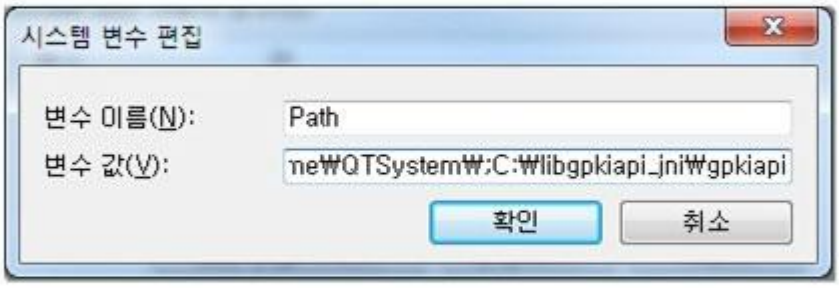
```
<sqlMapresource="egovframework/sqlmap/hyb/add/dvc/EgovPKIAndroidAPIGuide_SQL_[DB NAME].xml"/>
```

- Standard API setting

Category	**How to Configure**
Configuring Class Directory	Alt. 1 Use JavaScript : java -classpath jar_directory\libgpkiapi_jni.jar Alt 2. Register Environmental Variables: My Computer → Property → Advanced → Environmental Variables, select Create New and register Classpath Variables



Category	How to Configure
Configuring Library Directory	1. Make sure the environmental variable has the directory of the Standard API for Security and LDAP Library for C/C++, including JNI. 2. Go My Computer → Property → Advanced → Environmental Variables and add the library directory to the variable “path”.



Functions

NPKI Device API guide is comprised of **Select/authenticate mobile device Certificate** , **View authentication log** functions.

Select/authenticate mobile device Certificate

Business Logic

Inquires the list of certificates saved on the mobile device through Device API. Authenticates selected Certificate from the list.

Code

Inquires the list of Certificates through JavaScript code that uses the inquiry function within the Device API. Signs using the JavaScript that creates signature data.

```
// inquire the list of Certificates
function fn_egov_go_certlist() {
	console.log('DeviceAPIGuide fn_egov_go_certlist');
	WizSignPG.getCertificates(fn_egov_getcertlistSuccess,fn_egov_getcertlistFail);
}

// verifies Certificate password
function fn_egov_confirm_password() {
	console.log('DeviceAPIGuide fn_egov_confirm_password');

	var args = new Array();
	args[0] = Number(document.getElementById("xsigncertindex").value) + 1
	args[1] = $("#loginPasswd").val();

	WizSignPG.verifyCertPassword(args, function(result) {
		console.log('DeviceAPIGuide fn_egov_confirm_password Success');
		//Certificates
		var runResult = result['result'];
		var error = result['errMsg'];

		if(error!=null){
			alert(error);
			alert(runResult);
		}

		if(runResult == 'OK') {
			alert('Correct Password.');
			fn_egov_make_sign();
		}

		}, function(error) {
			console.log('DeviceAPIGuide fn_egov_confirm_password Fail');
			alert("Error: \r\n" + error['errMsg']);
	});
}

// signs the Certificate
function fn_egov_make_sign()
{
	console.log('DeviceAPIGuide fn_egov_make_sign');
	var args = new Array();
	args[0] = Number(document.getElementById("xsigncertindex").value) + 1
	args[1] = $("#loginPasswd").val();
	args[2] = "usrId=&password=&name=";

	WizSignPG.doSignature(args, fn_egov_makesign_ok, fn_egov_makesign_fail);
}

// requests authentication to Certificate signature data server
function fn_egov_makesign_ok(arg)
{
	console.log('DeviceAPIGuide fn_egov_makesign_ok Success');
	var jsonobj = JSON.parse(arg);
```

```
	// The signature value can be imported from jsonobj.sign and the VID Random value for
identification can be imported from jsonobj.vidRandom.
	var signedData = jsonobj.sign;

	var url = "/pki/xml/addPKIInfo.do";
	var acceptType = "xml";
	var params = {uuid :    device.uuid,
				sign: signedData,
				entrprsSeCode: 'PKI02'};
	alert('Http Method:POST\nacceptType:'+ acceptType + '\n Request Data:' +
JSON.stringify(params));

	// get the data from server
	window.plugins.EgovInterface.post(url,acceptType, params, function(xmldata) {
		console.log('DeviceAPIGuide fn_egov_makesign_ok request Complete');
		alert('Response Data:' + xmldata)
		if($(xmldata).find("resultState").text() == "OK"){
			window.history.go(-2);
		}else{
			jAlert($(xmldata).find("resultMessage").text(), 'Error', 'c');
		}
	});
}
```

Related Screen and Implementation Manual

Action	URL	Controller method	QueryID
Certificate authentication	/pki/xml/addPKIInfo.do	addPKIInfoXml	“PKIAndroidAPIDAO.insertPKIInfo”

Certificate list **Certificate authentication**

Select the Certificate to be authenticated from the Certificate list window. Enter the password on the password section of the authentication window, and click the "confirm" button.
An error message will be displayed if conditions are insufficient upon checking validation on the password section.

Confirm authentication: enter the Certificate password on the password section adn click "confirm" button.
Back button : moves to **NPKI Device API Guide Program menu** window or**Certificate list** window.

View authentication log

Business Logic

Updates the Certificate Authorization Log out of the web server application.

Code

```
function fn_egov_go_loginInfoList() {
	console.log('DeviceAPIGuide fn_egov_go_loginInfoList');
	// displays the warning message that data charges will be incurred when using 3G.
	if(!fn_egov_network_check(false)) {
		return;
	}

	$.mobile.changePage("#loginInfoList", "slide", false, false);

	var url = "/pki/xml/pkiInfoList.do";
	var accept_type = "xml";
	// get the data from server
	window.plugins.EgovInterface.post(url,accept_type, null, function(xmldata) {
		console.log('DeviceAPIGuide fn_egov_go_loginInfoList request Complete');
		var list_html = "";
		$(xmldata).find("pkiInfoList").each(function(){
			var dn = $(this).find("dn").text();
			var date = $(this).find("crtfcDt").text();
			var entrprsSeCode = $(this).find("entrprsSeCode").text().replace(/\s+$/, "");
			var entrprsSe = "NONE";
			if(entrprsSeCode == 'PKI01')
				entrprsSe = "MagicXSign";
			else if(entrprsSeCode == 'PKI02')
				entrprsSe = "WizSign";
			else if(entrprsSeCode == 'PKI03')
				entrprsSe = "XecureSmart";

			list_html += "<li><h3>subjdn : " + dn + "</h3>";
			list_html += "<p><strong>Date : " + date + "</strong></p>";
			list_html += "<p><strong>NPKI : " + entrprsSe + "</strong></p></li>";
		});
		var theList = $('#theLogList');
		theList.html(list_html);
		theList.listview("refresh");
		setTimeout(loadiScrollList, 1000);
	});
}
```

Related Screen and Implementation Manual

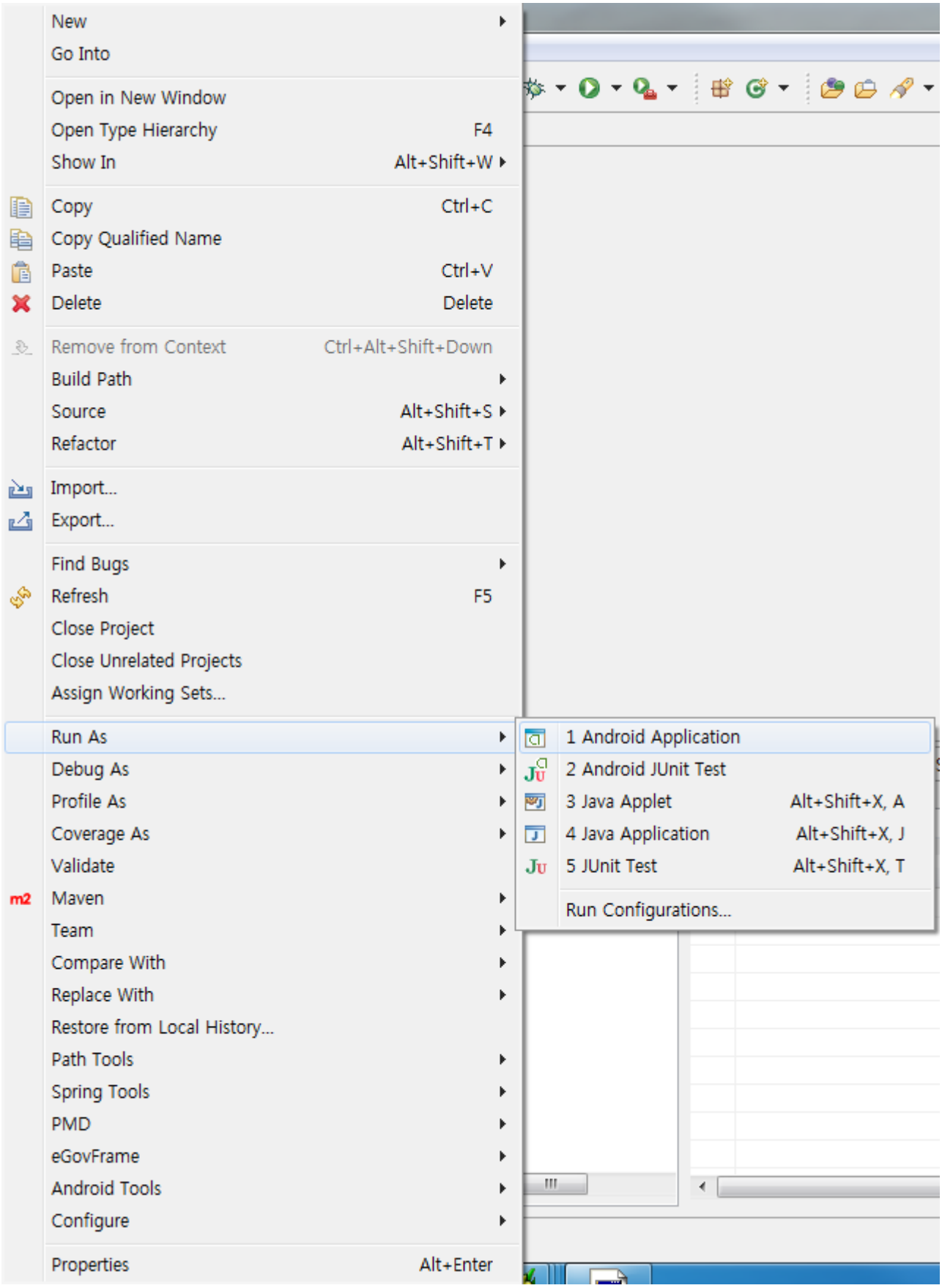
Function	URL	Controller	method	QueryID
Inquire Certificate authentication results log	/pki/xml/pkiInfoList.do	EgovPKIAndroidAPIController	selectPKIInfoListXml	PKIAndroidAPIDAO.selectPKIInfoList

Compiling, debugging, distributing

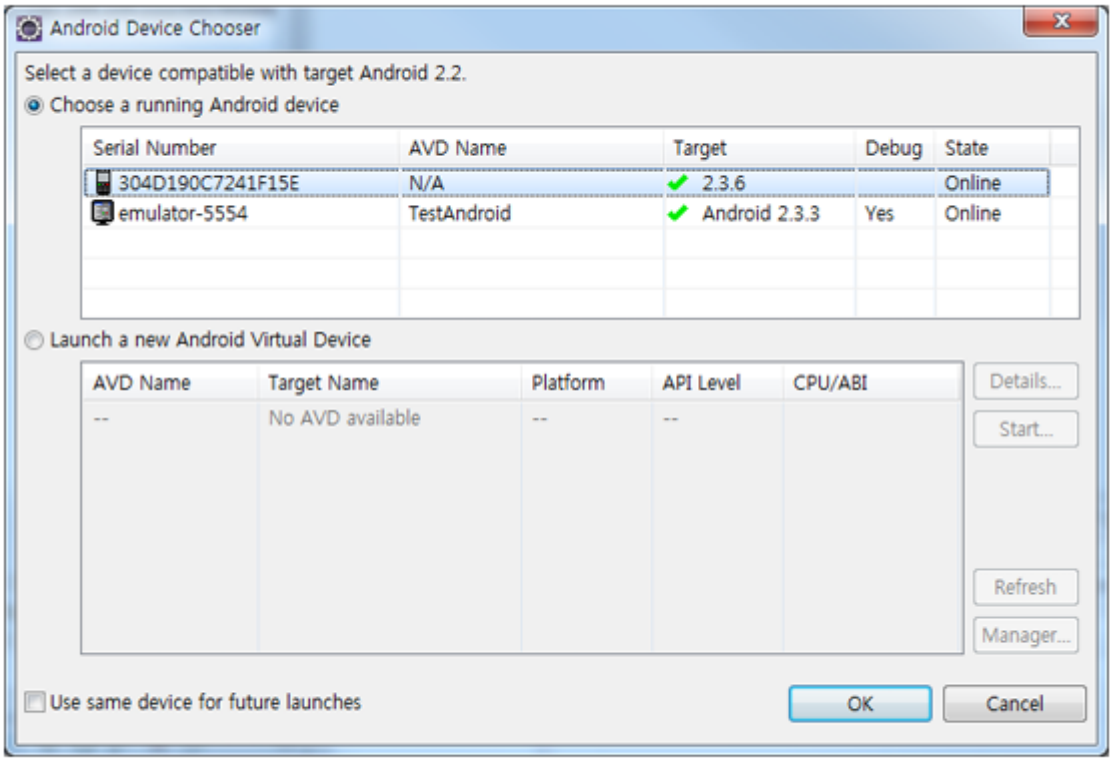
Compiling

How to compile NPKIAPI(WizSign) Device Applicaton

- Right-click on the Device API Guide(Android) project, and click on the "Android Application" at the "Run As" tab. The guide program will be built and installed into the Android device.



- When “Android Device Chooser” window appears, select appropriate device and click on the "OK" button.



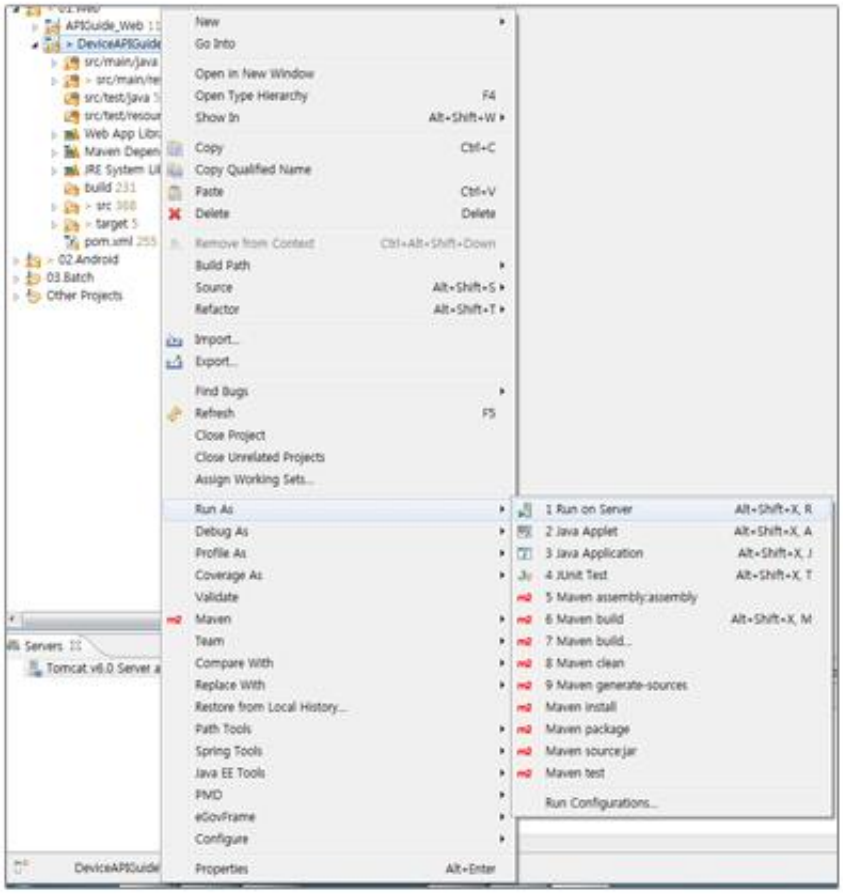
- Program display on the emulator

- Program display on the device


olleh
오후 1:57
100%
PKI(WizSign) API 디바이스
가이드 프로그램
행정안전부

How to compile NPKIAPI(WizSign) Server Applicaton

- Right-click on the project and click on Run As>Run On Server in order to run the NPKIAPI(WizSign) server-side Guide Program.



- When the build is successfully completed, a message reading 'Server Startup in xxx ms' will display on the console window on the Eclipse.

```
2012-09-14 09:15:49,759 DEBUG [org.springframework.beans.factory.support.DefaultListableBeanFactory] Returning cached instance of singleton bean 'org.springframework.web.serv
2012-09-14 09:15:49,767 DEBUG [org.springframework.beans.factory.support.DefaultListableBeanFactory] Returning cached instance of singleton bean 'org.springframework.web.serv
2012-09-14 09:15:49,768 DEBUG [org.springframework.beans.factory.support.DefaultListableBeanFactory] Creating instance of bean 'org.springframework.web.servlet.view.DefaultRe
2012-09-14 09:15:49,771 DEBUG [org.springframework.beans.factory.support.DefaultListableBeanFactory] Finished creating instance of bean 'org.springframework.web.servlet.view.
2012-09-14 09:15:49,771 DEBUG [org.springframework.web.servlet.DispatcherServlet] Unable to locate RequestToViewNameTranslator with name 'viewNameTranslator': using default
2012-09-14 09:15:49,771 DEBUG [org.springframework.beans.factory.support.DefaultListableBeanFactory] Returning cached instance of singleton bean 'org.springframework.web.serv
2012-09-14 09:15:49,771 DEBUG [org.springframework.beans.factory.support.DefaultListableBeanFactory] Returning cached instance of singleton bean 'viewResolver'
2012-09-14 09:15:49,772 DEBUG [org.springframework.web.servlet.DispatcherServlet] Published WebApplicationContext of servlet 'action' as ServletContext attribute with name [o
2012-09-14 09:15:49,772  INFO [org.springframework.web.servlet.DispatcherServlet] FrameworkServlet 'action': initialization completed in 1373 ms
2012-09-14 09:15:49,772 DEBUG [org.springframework.web.servlet.DispatcherServlet] Servlet 'action' configured successfully
2012. 9. 14 오전 9:15:49 org.apache.coyote.http11.Http11Protocol start
정보: Starting Coyote HTTP/1.1 on http-80
2012. 9. 14 오전 9:15:49 org.apache.jk.common.ChannelSocket init
정보: JK: ajp13 listening on /0.0.0.0:8009
2012. 9. 14 오전 9:15:49 org.apache.jk.server.JkMain start
정보: Jk running ID=0 time=0/30  config=null
2012. 9. 14 오전 9:15:49 org.apache.catalina.startup.Catalina start
정보: Server startup in 7209 ms
```

Debugging

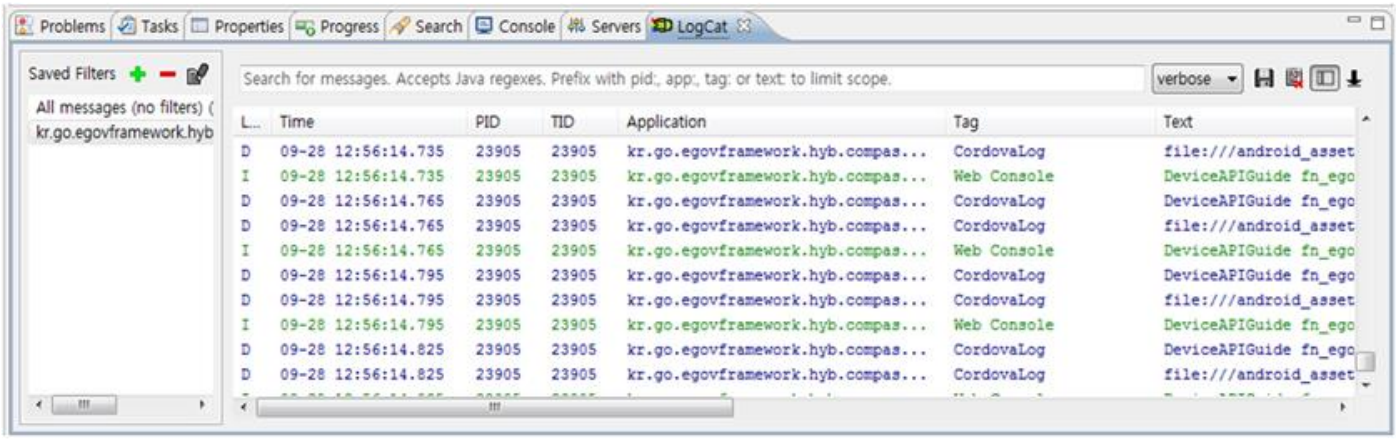
Use console.log in order to check the details on any errors on the device application, and to conduct debugging. Debug codes in console.log are available in JavaScript syntaxes that you can use in Eclipse.

See the following for how to code console.log:

```
function fn_egov_getcertlistSuccess(result)
{
    console.log('DeviceAPIGuide fn_egov_getcertlistSuccess Success');
```

```
        g_certList = result['Certificates'];
        var list_html = "";
        ....
}
```

When the debugging code is executed, check out the following console message appears:



NPKI device API Guide Program will output the following console information for debugging.

Debug code	Debug information
DeviceAPIGuide fn_egov_getcertlistSuccess Success	Certificate list inquiry successful
DeviceAPIGuide fn_egov_getcertlistFail Fail	Certificate list inquiry failed
DeviceAPIGuide fn_egov_confirm_password Success	Certificate password verification successful
DeviceAPIGuide fn_egov_confirm_password Fail	Certificate password verification failed
DeviceAPIGuide fn_egov_makesign_ok Success	Certificate signing successful
DeviceAPIGuide fn_egov_makesign_fail Fail	Certificate signing failed
DeviceAPIGuide fn_egov_makesign_ok request Complete	Certificate authentication from web server application succ essful
DeviceAPIGuide fn_egov_go_loginInfoList request Comp lete	Certificate authentication log information inquiry successful

Distribution

Download NPKI(WizSign) Device API guide : Click

References

- UX/UI library : jQuery MobileClick
- Phonegap 4.3.0 : Click
- NPKIAPI : KSign Inc. http://www.ksign.com

Standard Security API : Click