NPKI(XecureSmart) DeviceAPI Guide Program

Outline

NPKI(XecureSmart) 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	Xcode 6.3.2, PhoneGap 4.3.0
Server-side Developmental Environment	eGov Standard Framework Develeopment Environment 3.5
Works in sync with Mash up Open API	N/A
Test Device	iPhone4, iPhone6
Test Platform	iOS 7.1.2, iOS 8.3
Libraries Added	XecureSmart library applied wizsXSCore.js, libKeySharpiPhone.a, libXecureSmartPG.a, XecureSmartPG.h, XWUIErrorCode.h, CopyCertificate.h, libXecureSmartPG.a, bridge, crypto

Restriction

NPKI library

- Since NPKI Device API Guide Program does not include security module, one must request license contract and support to security module firm below.

Name	Point of Contact	Contact Phone	Homepage
Softforum Inc.	Shin, Min-Gyu	031-622-6300	http://www.softforum.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

For iPhone devices, there may be issues due to device's processing power.

- Problem: PhoneGap error.
- Solution: delay PhoneGap loading sequence with setTimeout() function.

```
document.addEventListener('DOMContentLoaded', function () { setTimeout(loaded, 200); }, false);
```

- Problem: iScroll5 content height calculation error.
- Solution: use setTimeout() to ensure iscroll is generated after css application to contents is complete.

```
setTimeout(function()
    {
        myScroll = new iScroll(thisPage,
            {
                checkDOMChanges: true,
                onBeforeScrollStart:function(e)
```

```
                              {
                              }
                        });
      },
      500);
      Problems may occur if alert() is included in CallBack function. (phoneGap)
```

- Problem: error calling alert() message from CallBack function saved in PhoneGap.
- Solution: Use asynchronous function or avoid using functions that use thread like alert().
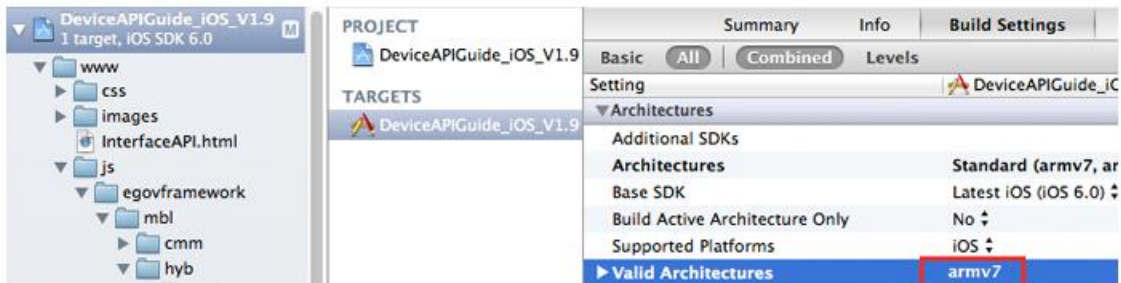
Caution when executing in Xcode 4.5

iOS 6, iPhone 5 is supported from Xcode 4.5. When supporting this device and OS, project options must be changed since PhoneGap 1.9.0 library is distributed after being compiled in Xcode 4.X.

1. Option setting


2. delete armv7s

Using cross domain

When using certain outside domains or its subdomains on PhoneGap, add such domains on <access origin="ExternalHosts"/> at [Project_Name]/Supproting Files/config.xml

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)

Class Diagram
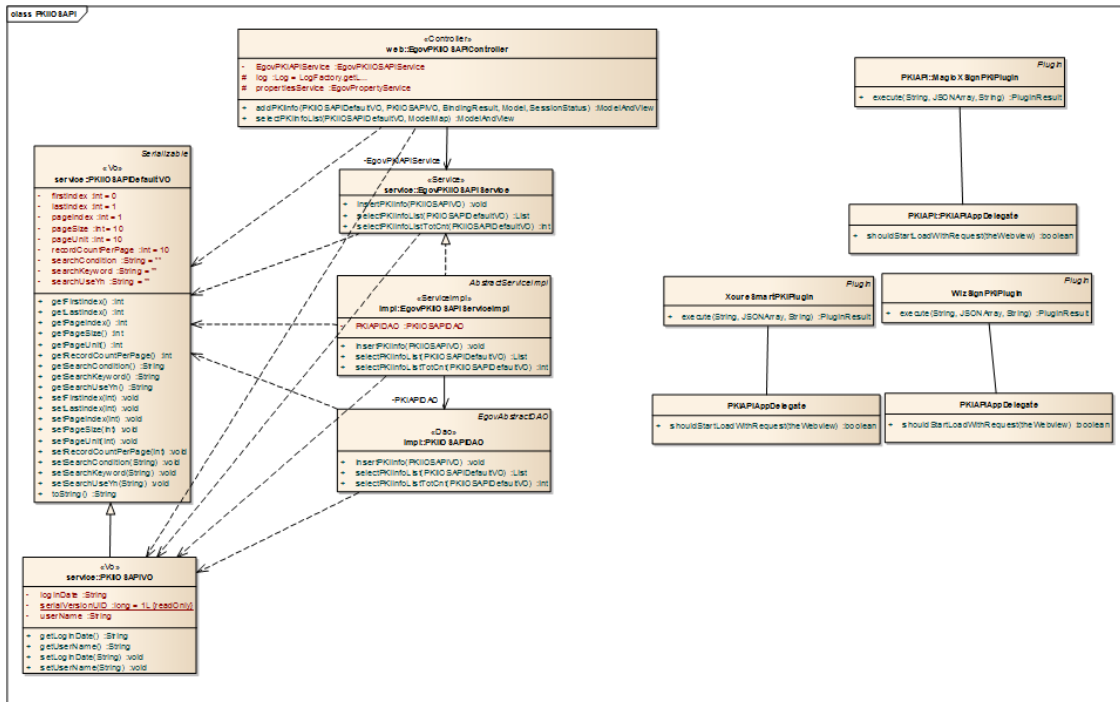
Device Application

Source

Type	Title	Remark
CSS	www/css/egovframwork/mbl/hyb/PKIXecureSmartAPI.css	NPKIAPI Guide Program Core Cascading Style Sheets
IMAGE	www/images/egovframwork/mbl/hyb/	NPKIAPI Guide Program main Image Folder
JS	www/js/egovframwork/mbl/hyb/PKIXSCoreAPI.js	NPKIAPI Guide Program main JavaScript
JS	www/js/egovframwork/mbl/hyb/XSCore.js	NPKIAPI Guide Program main JavaScript
JS	www/js/egovframwork/mbl/hyb/messages_ko.js	JavaScript for Validate Message Processing
HTML	www/NPKIXecureSmartAPI.html	NPKIAPI main page
HTML	www/license.html	NPKIAPI license page
HTML	www/overview.html	NPKIAPI feature description page

Function API

XecureSmart API DOC

APIs Used

XecureSmartPlugin.getCertTree

- Calls device's Certificate list. Search can be done using both searchType and searchValue, and also with searchSerial for serial numbers.

void XecureSmartPlugin.getCertTree (successCB , failCB , certType , searchType , contentLevel , searchValue , searchSerial)

Option	Description	Remark
successCB	Callback function when successful	(out)contentLevel : 0(detailed), contentLevel : 5(simple)
failCB	Callback function when failed	(out)error code$error message
certType	Type	(in)0:root Certificate,1:CACertificate,2:user Certificate,3: all Certificate
searchType	Search condition	(in)0:do not search 10:subjectDN CN match 11:subjectDN OU match 12:subjectDN O match 13:subjectDN C match 14:subjectDN match 20:issuerDN CN match 21:issuerDN OU match 22:issuerDN O match 23:issuerDN C match 24:issuerDN match
contentLevel	Result value's level	(in)0:detailed info,5:simple info
searchValue	Search value	(in)
searchSerial	Search Serial No.	(in)

XecureSmartPlugin.signDataCMS

- Electronically signs plain text.

void XecureSmartPlugin.signDataCMS (successCB , failCB , issuerDN , serial , password , plainText)

Option	Description	Remark
successCB	Callback function when successful	(out)Electronic signature text
failCB	Callback function when failed	(out)error code$error message
issuerDN	Certificate issuer	(in)
serial	Certificate Serial No.	(in

password	Certificate password	(in)
plainText	Plain text	(in)

Server Application

Source

| Type	Title	Remark
Controller	egovframework.hyb.ios.pki.web.EgovPKIiOSAPIController.java	NPKIAPI Guide Program Controller Class
Service	egovframework.hyb.ios.pki.service.EgovPKIiOSAPIService.java	NPKIAPI Guide Program Service Class
ServiceIimpl	egovframework.hyb.ios.pki.service.impl.EgovPKIiOSAPIServiceImpl.java	NPKIAPI Guide Program ServiceImpl Class
VO	egovframework.hyb.ios.pki.service.PKIiOSAPIDefaultVO.java	NPKIAPI Guide Program VO Class
VO	egovframework.hyb.ios.pki.service.PKIiOSAPIVO.java	NPKIAPI Guide Program VO Class
VO	egovframework.hyb.ios.pki.service.PKIiOSAPIXmlVO.java	NPKIAPI Guide Program XML related VO Class
DAO	egovframework.hyb.ios.pki.service.impl.PKIiOSAPIDAO.java	NPKIAPI Guide Program Dao Class
QUERY XML	resources/egovframework/sqlmap/hyb/ios/pki/EgovPKIiOSAPIGuide_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 authentication Log Management

Table Definition

- PKI

No	Column ID	Title of Column	Type	Length	Null
1	SN	Serial No.	NUMERIC	6	NotNull
2	UUID	UUID	VARCHAR	50	NotNull
3	DN	Authentication data	VARCHAR	255	Null
4	CFTFC_DT	Authentication date and time	DATETIME		Null
5	ENTRPRS_SE_CODE	Enterprise code	CHAR	15	Null

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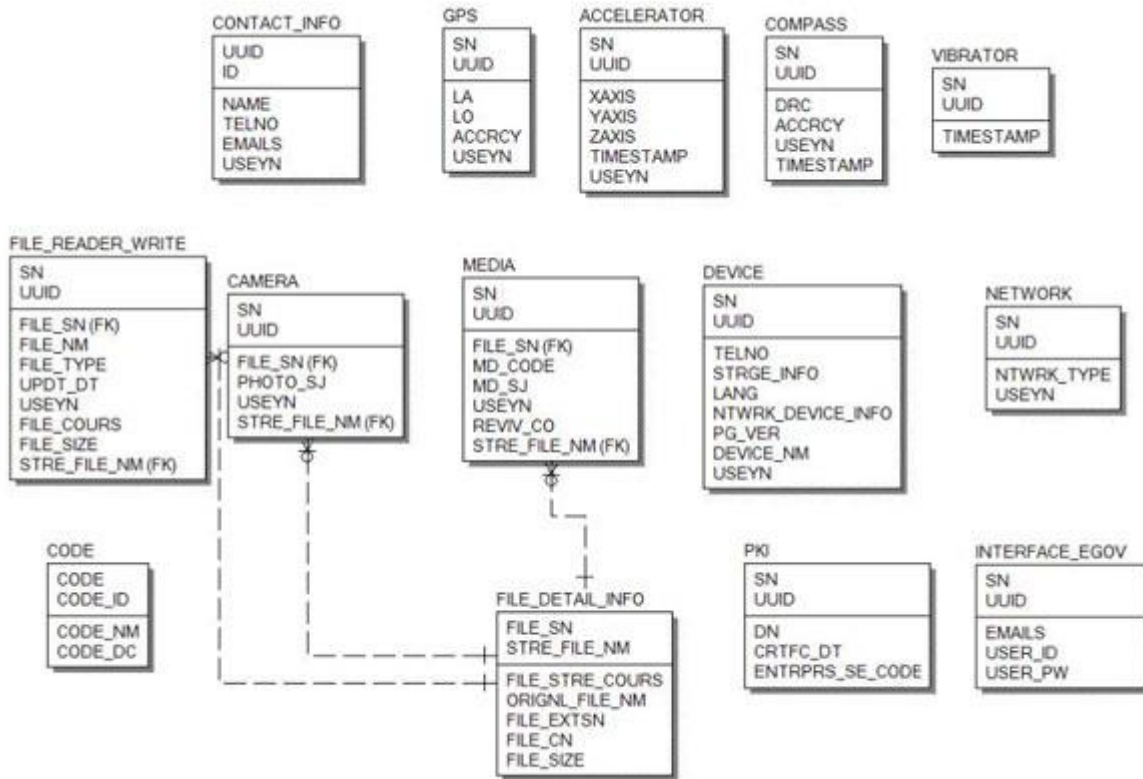
```java
public String verifyCert(PKIAndroidAPIVO pkiVo) throws Exception {
    // API initialization
    GpkiApi.init("C:/libgpkiapi_jni/conf");
    String sign;
    sign = pkiVo.getSign();
    return verify(Base64.decode(sign));
}

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;
}
```

Configuration Settings

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

Device Application

config.xml

Plugin

```
<featurename="InterfaceAPI">
<paramname="ios-package"value="EgovInterface"/>
</feature>
```
[Project_Name]/eGovModule/EGovComModule.h

```
<!-- Server Directory for eGov Interface Device API Class -->
#define kSERVER_URL        @"Server_URL"
```
Server Application

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

```
<sqlMapresource="egovframework/sqlmap/hyb/ios/pki/EgovPKIiOSAPIGuide_SQL_[DB
NAME].xml"/>
```

Setting reference

Related features

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.

Related 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');
    $.mobile.showPageLoadingMsg('a');
    XecureSmart.getCertTree(fn_egov_getcerttree_success, fn_egov_getcerttree_fail, 2, 0, 5, '', '');
}
```

```
// signs the Certificate
function fn_egov_make_sign()
{
    console.log('PKIXSCoreAPIGuide fn_egov_make_sign');
    XecureSmart.signDataCMS (fn_egov_makesign_ok, fn_egov_makesign_fail,
document.getElementById("issuerDN").value, document.getElementById("certSerial").value,
$("#loginPasswd").val(), "usrId=&password=&name=");
}
```

```
// requests authentication to Certificate signature data server
function signDataCMS_success (result)
{
    console.log("PKIXSCoreAPIGuide signDataCMS_success Success");
    console.log('result : ' + result);
    if ('eSign Failed.' == result)
    {
        alert('eSign Failed.');
        return;
    }

    //    var signedData = arg['signedData'];
    var params = {uuid :   device.uuid,
                      sign: result,
                      entrprsSeCode: 'PKI03'};
    $.mobile.showPageLoadingMsg('a');
```

```
alert('Http Method:POST\nAcceptType:JSON\nSend Data:' + JSON.stringify(params));

EgovInterface.submitAsynchronous(
                                    [params, "/pki/addPKIiOSInfo.do"],
                                    function(result) {
                                        console.log("PKIXSCoreAPIGuide
signDataCMS_success request Completed");

                                        var str = '{';
                                        for (myKey in result){
                                            str += myKey + ':' + result[myKey] + '\n';
                                        }
                                        str += '}';
                                        alert('Response
Method:RESTful\nResponseType:json, post\nParam:\n' + str);
                                        //window.history.back();
                                        $.mobile.hidePageLoadingMsg('a');
                                        location.href = "index.html";
                                    },
                                    function(error) {
                                        console.log("PKIXSCoreAPIGuide
signDataCMS_success request Failed");

                                        var str = '{';
                                        for (myKey in error){
                                            str += myKey + ' : ' + error[myKey] + '\n';
                                        }
                                        str += '}';
                                        alert('Response Method:RESTful\nSendType:json,
post\nParam:\n' + str);

                                        $.mobile.hidePageLoadingMsg('a');
                                    }
                                    );

}
```

Related Screen and Implementation Manual

Action	URL	Controller method	QueryID
Certificate authentication	/pki/xml/addPKIiOSInfo.do	addPKIInfoXml	"PKIiOSAPIDAO.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.

Related Code

```
function fn_egov_go_loginInfoList()
{
    $.mobile.changePage("#loginInfoList", "slide", false, false);

    // get the data from server
    console.log('fn_egov_go_loginInfoList()');
    $.mobile.showPageLoadingMsg('a');
    var accept_type = "json";
    // get the data from server
    EgovInterface.submitAsynchronous(
                                    ["/pki/pkiInfoList.do"],
                                    function(result) {
                                        console.log("PKIXSCoreAPIGuide
fn_egov_go_loginInfoList request Completed");

                                        var list_html = "";
                                        var totcnt = result.pkiInfoList.length;

                                        for (var i = 0; i < totcnt; i++) {
                                            var data = result.pkiInfoList[i];
                                            var entrprsSe = "NONE";
                                            var entrprsSeCode = data.entrprsSeCode;
                                            if(entrprsSeCode == 'PKI01')
                                                entrprsSe = "MagicXSign";
                                            else if(entrprsSeCode == 'PKI02')
                                                entrprsSe = "WizSign";
                                            else if(entrprsSeCode == 'PKI03')
                                                entrprsSe = "XecureSmart";

                                            list_html += "<li><h3>subjdn : " + data.dn +
"</h3>";

                                            list_html += "<p><strong>Date : " +
data.crtfcDt + "</strong></p>";

                                            list_html += "<p>NPKI : " + entrprsSe +
"</p></li>";

                                        }
                                        var theList = $('#theLogList');
                                        theList.html(list_html);
                                        $.mobile.changePage("#loginInfoList", "slide",
false, false);

                                        theList.listview("refresh");
                                        $.mobile.hidePageLoadingMsg('a');
                                        setTimeout(loadiScrollList, 1000);
                                    },
                                    function(error) {
                                        console.log("PKIXSCoreAPIGuide
fn_egov_go_loginInfoList request Failed");

                                        var str = '{';
                                        for (var myKey in error){
                                            str += myKey + ' : ' + error[myKey] + '\n';
                                        }
                                        str += '}';
```

```
                                        alert('Response Method:RESTful\nSendType:json,
post\nParam:\n' + str);

                                        $.mobile.hidePageLoadingMsg('a');
                        }
                    );
}
```

Related Screen and Implementation Manual

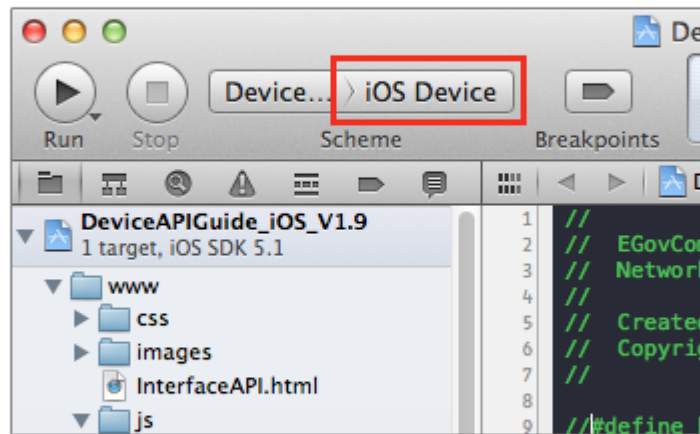| Function	URL	Controller	method	QueryID
Inquire Certificate authentication results log	/pki/xml/pkiInfoList.do	EgovPKIiOSAPIController	selectPKIInfoListXml	PKIiOSAPIDAO.selectPKIInfoList

Compiling, debugging, distributing

Compiling

How to compile NPKI Device Applicaton

1. To execute on the device or simulator, click on red border area.

2. Select device or simulator.

3. Click on "Execute."
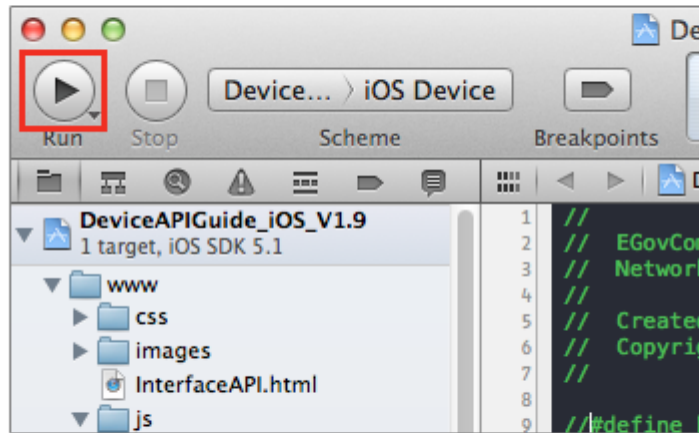
4. Check intro and main screen.

How to compile NPKI Server Applicaton

- Right-click on the project and click on Run As>Run On Server in order to run the NPKIAPI server-side Guide Program.
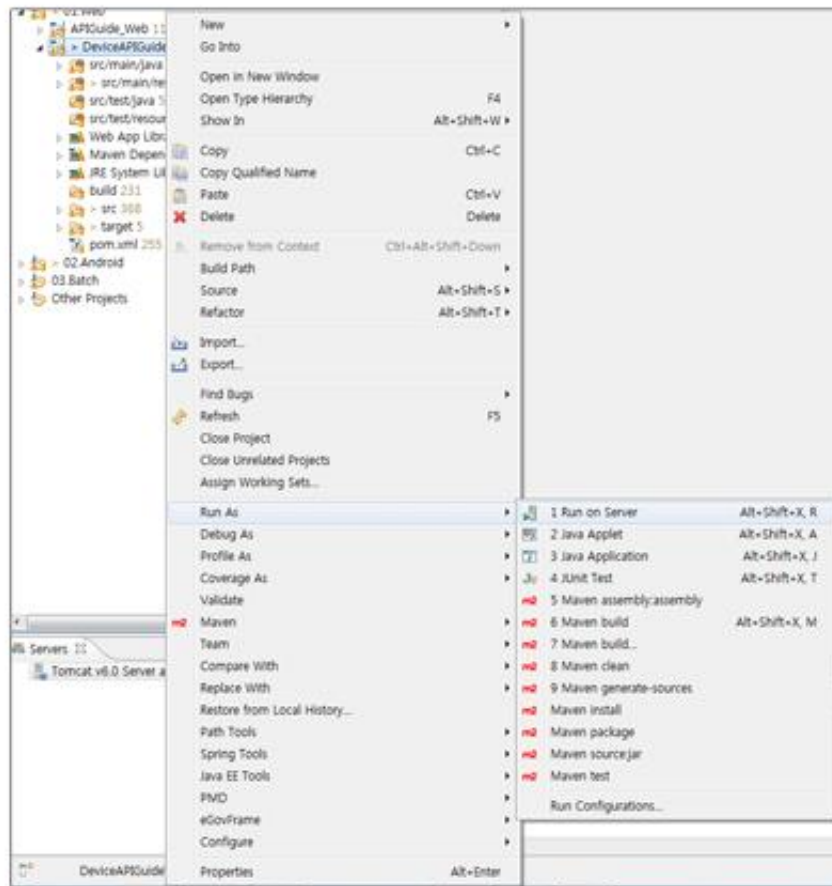
- When the build is successfully completed, a message reading 'Server Startup in xxx ms' will display on the console window on the Eclipse.
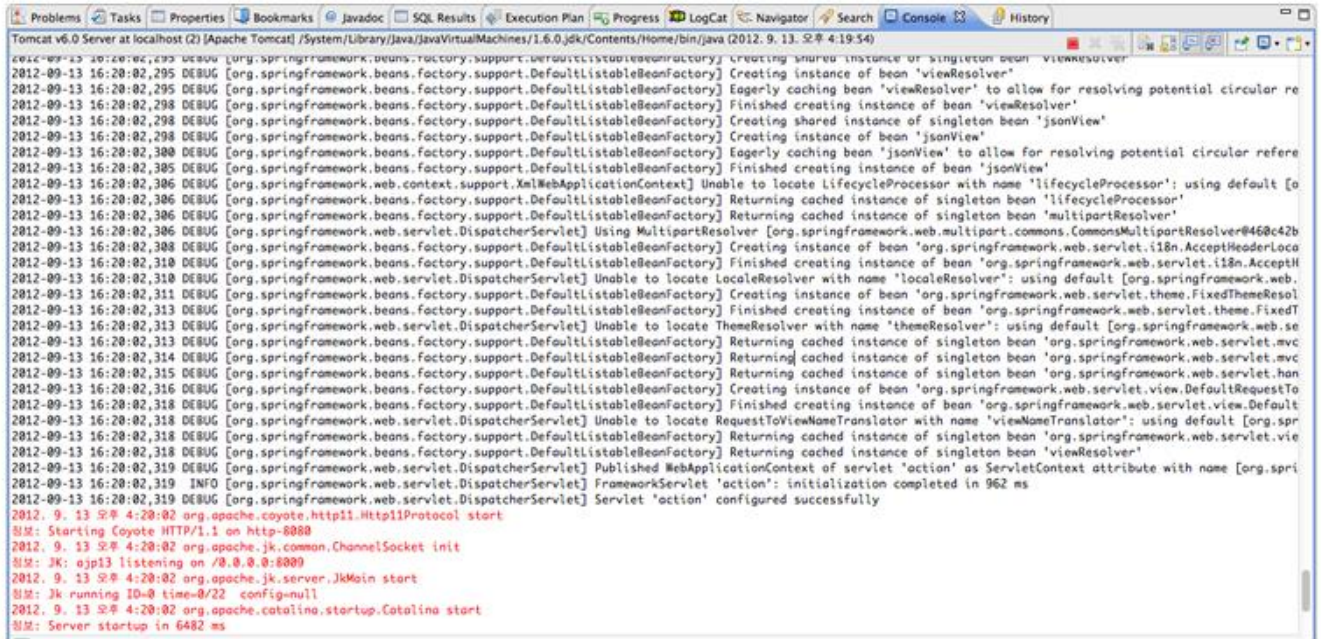
Debugging

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 both Eclipse and Xcode.

- Example of actual console log

```
function fn_egov_network_check(doCheck)
{
    console.log('DeviceAPIGuide fn_egov_network_check');
    var networkState = navigator.network.connection.type;
    ...
}
```

- xCode console window

- Organizer log window

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

Debug code	Debug information
PKIXSCoreAPIGuide deviceready Success	Device ready complete
PKIXSCoreAPIGuide getCertTree_success Success	Certificate loading successful
PKIXSCoreAPIGuide getCertTree_fail Failed	Certificate loading failed
PKIXSCoreAPIGuide signDataCMS_success Success	Signing value loading successful

PKIXSCoreAPIGuide signDataCMS_fail Failed   Signing value loading failed

PKIXSCoreAPIGuide signDataCMS_success request Completed   Transfer of signed value to server successful

PKIXSCoreAPIGuide signDataCMS_success request Failed   Transfer of signed value to server failed

PKIXSCoreAPIGuide fn_egov_go_loginInfoList request Completed Loading authentication list from server successful

PKIXSCoreAPIGuide fn_egov_go_loginInfoList request Failed   Loading authentication list from server failed

    Distribution

Download NPKI Device API Guide: [Click]

References

- UX/UI library : jQuery Mobile[Click]
- Phonegap 4.3.0 : [Click]
- Standard security API : http://www.gpki.go.kr/