

Device Info Device API Guide Program

Outline

Device Info guide program 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 inquiry of meta information of mobile smart devices through JavaScript-based DeviceInfoAPI. Also, it is comprised of a feature to send and inquire meta information within the Device, in conjunction with eGovFramework based web server application.

Feature

This Guide Program provides **device information inquiry, send device information to server, receive device information from server** functions.

Preconditions

Category	Description
Local Device Environments	eGovFramework Runtime Environment 3.5, Android SDKAPI 22(version 5.0 Lollipop)
Server-side Developmental Environment	Runtime Environment for eGov Standard Framework 3.5
Works in sync with Mash up Open API	N/A
Test Device	Galaxy S2
Test Platform	Android 2.3
Libraries Added	N/A

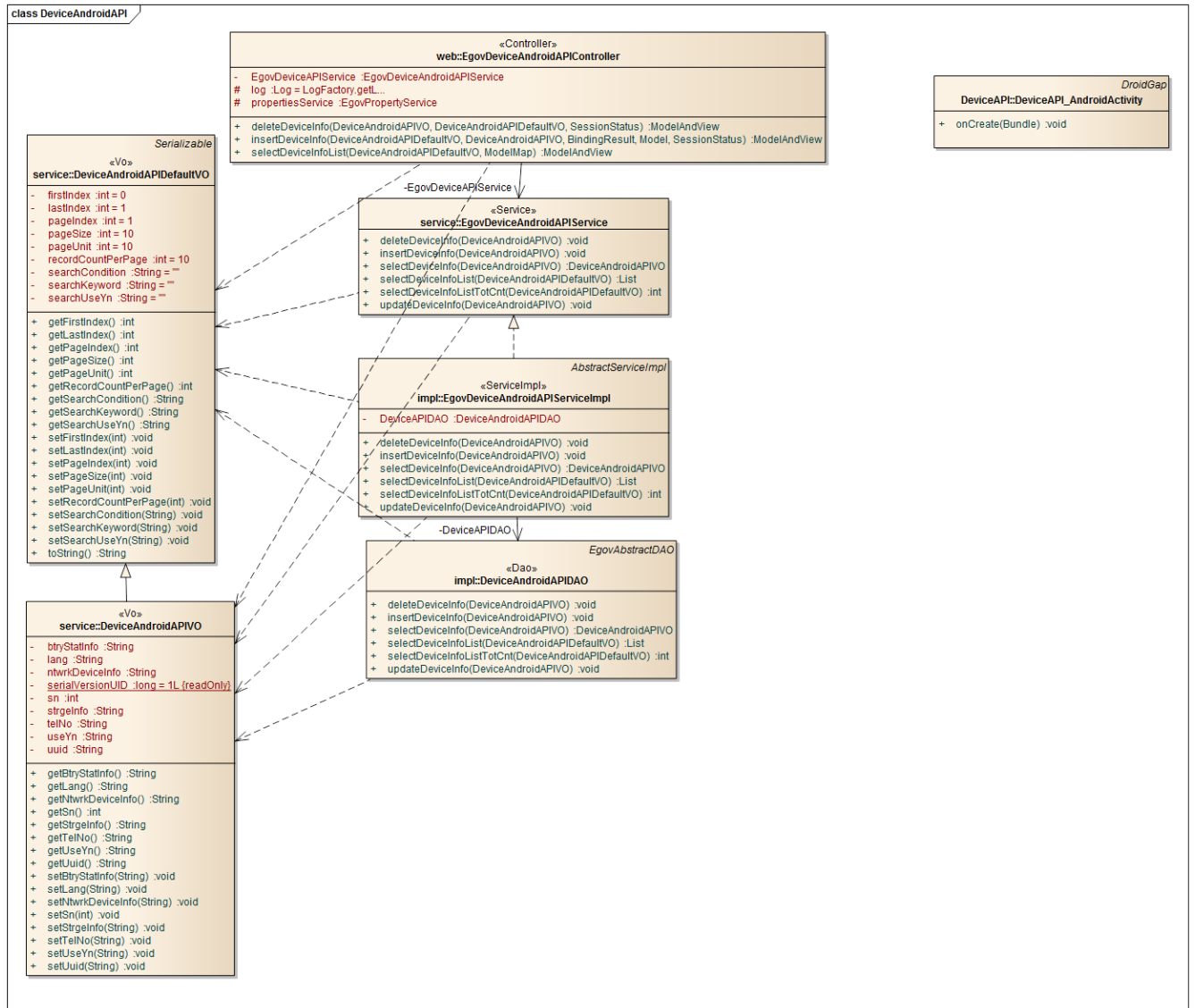
Restriction

Category	Description
Supported devices and platforms	N/A
Using cross domain	When using certain outside domains or its subdomains on PhoneGap, add such domains on <code><access origin="" /></code> at Res/xml/config.xml.
License	N/A

Description

DeviceInfo Device API Guide Program is comprised of **device information inquiry**, **send device information to server**, and **receive device information from server** features.(refer to related features section)

Class Diagram



Device Application

Source

Type	Title	Remark
Activitiy	kr.go.egovframework.hyb.DeviceInfoapi.DeviceInfoAPIGuide_ AndroidActivity	DeviceInfoAPI Guide Program Activity Class
CSS	assets/www/css/egovframework/mbl/hyb/DeviceInfoAPI.css	DeviceInfoAPI Guide Program main Cascading Style Sheets

IMAGE	assets/www/images/egovframework/mbl/hyb/	DeviceInfoAPI Guide Program main Image folder
JS	assets/www/js/egovframework/mbl/hyb/DeviceInfoAPI.js	DeviceInfoAPI Guide Program main JavaScript
JS	assets/www/js/egovframework/mbl/hyb/messages_ko.js	JavaScript for Validate Message Processing
RES	assets/www/res/	DeviceInfoAPI Guide Program main Resource folder
XML	AndroidManifest.xml	Configuration XML for Android
HTML	assets/www/DeviceInfoAPI.html	DeviceInfoAPI Main Page
HTML	assets/www/intro.html	DeviceInfoAPI Intro Page
HTML	assets/www/license.html	DeviceInfoAPI License Page
HTML	assets/www/overview.html	DeviceInfoAPI Function Explanation Page

APIs Used

device

- Inquiry to meta information related to both hardware and software of the mobile device

```
var name = device.name;
var cordova= device.cordova;
var platform = device.platform;
var uuid = device.uuid;
var version = device.version;
```

Parameter	Description	Remark
name	Returns the pre-configured name	
cordova	Returns the PhoneGap version information	
platform	Returns the platform information of the mobile device	
uuid	Returns UUID of the mobile device	
version	Returns the platform version of the mobile device	

Event[batterycritical]

- Definition of Callback function that is called when battery is at critical status.

```
window.addEventListener("batterycritical", yourCallbackFunction, false);
Event[batterylow]
```

- Definition of Callback function that is called when battery is at low status.

```
window.addEventListener("batterylow", yourCallbackFunction, false);
```

Event[batterystatus]

- Definition of Callback function that is called when changes to battery status is detected.

```
window.addEventListener("batterystatus", yourCallbackFunction, false);
    eGov Custom Plug-In StorageInfo
```

- Returns storage capacity of mobile device.

```
var StorageInfo = {
    totalFileSystemSize: function(success, fail, types) {

        return Cordova.exec(success, fail, "StorageInfoPlugin", "fileSystemSize", types);
    }
}
    eGov Custom Plug-In DeviceNumber
```

- Returns the phone number of mobile device.

```
var DeviceNumber = {
    getDeviceNumber: function(success, fail, types) {

        return Cordova.exec(success, fail, "DeviceNumberPlugin", "deviceNumber", types);
    }
}
    Server Application
```

Source

Type	Title	Remark
Controller	egovframework.hyb.add.dvc.web.EgovDeviceAndroidAPIController.java	DeviceInfoAPI Guide Program Controller Class
Service	egovframework.hyb.add.dvc.service.EgovDeviceAndroidAPIService.java	DeviceInfoAPI Guide Program Service Class
ServiceImpl	egovframework.hyb.add.dvc.service.impl.EgovDeviceAndroidAPIServiceImpl.java	DeviceInfoAPI Guide Program ServiceImpl Class
VO	egovframework.hyb.add.dvc.service.DeviceAndroidAPIVO.java	DeviceInfoAPI Guide Program VO Class
DAO	egovframework.hyb.add.dvc.service.impl.DeviceAndroidAPIDAO.java	DeviceInfoAPI Guide Program Dao Class
QUERY ML	X resources/egovframework/sqlmap/hyb/add/dvc/EgovDeviceAndroidAPIGuide_SQL_XXX.xml	DeviceInfoAPI Guide Program QUERY XML

Related Tables

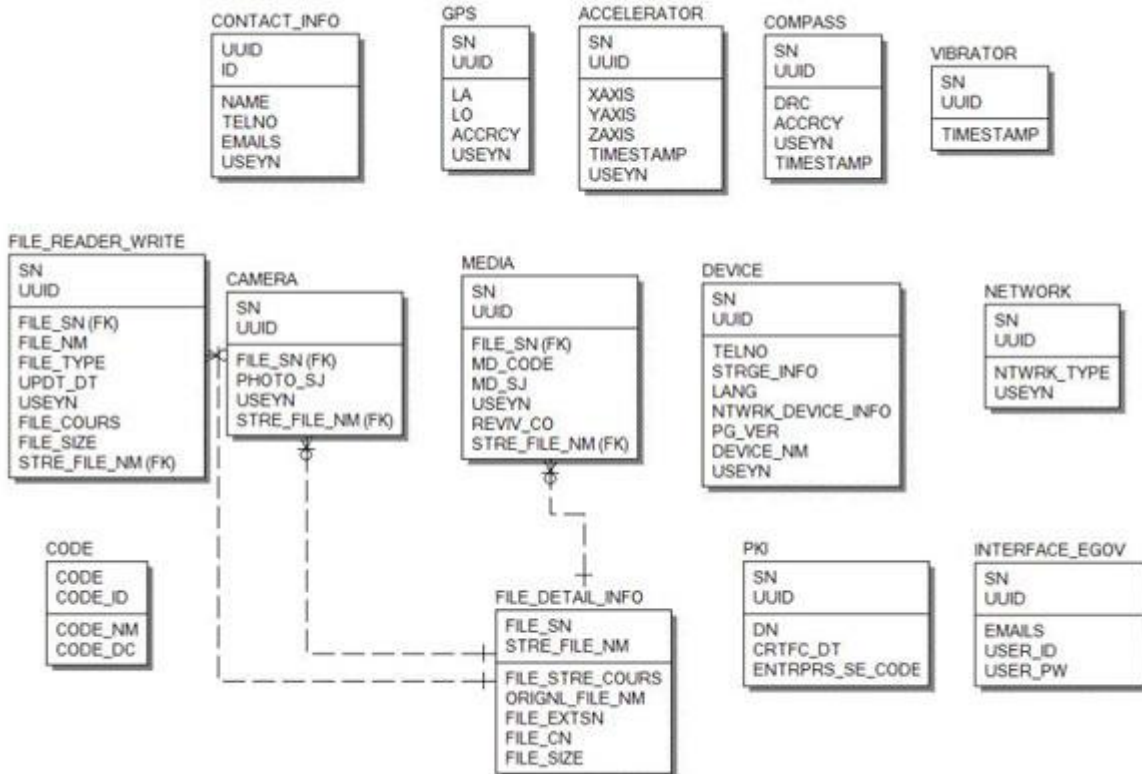
Title	Table	Remark
Device	Device Manages	Device information

Table Definition

- Device

No	Column ID	Title of Column	Type	Length	NULL
1	SN	Serial No.	NUMERIC	6	NotNull
2	UUID	UUID	VARCHAR	50	NotNull
3	TELNO	Phone contact	VARCHAR	20	Null
4	STRGE_INFO	Storage information	VARCHAR	40	Null
5	LANG	Language	VARCHAR	15	Null
6	NTWRK_DEVICE_INFO	Network device information	VARCHAR	20	Null
7	PG_VER	PhoneGap version information	VARCHAR	40	Null
8	DEVICE_NM	Device name	VARCHAR	200	Null
9	USEYN	Activation	CHAR	1	Null

ERD



Properties

Necessary sections and settings for using meta information inquiry feature of the mobile device, provided by DeviceInfo 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"/
>
```

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

context-properties.xml

- Upload directory

```
<entrykey="fileStorePath" value="File_Save_Path"/>
    resource/egovframework/sqlmap/sql-map-config_[DB_NAME].xml
```

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

Functions

DeviceInfo Device API Guide Program is comprised of features to a) inquire meta information within mobile device and b) send inquired meta information to server and inquire it.

device information inquiry

Business Logic

- Check the network. If the network is not Wi-Fi, function based on the user approval.

Related Codes

Device information VO

```
/** Device information VO */
var deviceInfoVO = {
    sn : "",
```

```

    uuid : "",
    os : "",
    telno : "",
    strgeInfo : "",
    ntwrkDeviceInfo : "",
    pgVer : "",
    deviceNm : "",
    useyn : ""
}

```

Displays device information screen

```

function fn_egov_load_deviceInfo() {
    deviceInfoVO.ntwrkDeviceInfo = fn_egov_get_networkInfo();

    $('deviceInfo:eq(0)').html(device.platform + " " + device.version);
    $('deviceInfo:eq(1)').html(device.uuid);
    $('deviceInfo:eq(2)').html(device.cordova);
    DeviceNumber.getDeviceNumber(fn_egov_deviceNumber_success, fn_egov_fail, []);
    $('deviceInfo:eq(4)').html(deviceInfoVO.ntwrkDeviceInfo);
    StorageInfo.totalFileSystemSize(fn_egov_totalSpace_success, fn_egov_fail, []);
}

```

Updates network information

```

function fn_egov_get_networkInfo() {
    var networkState = navigator.network.connection.type;

    var states = { };
    states[Connection.UNKNOWN] = 'Unknown connection';
    states[Connection.ETHERNET] = 'Ethernet connection';
    states[Connection.WIFI] = 'WiFi connection';
    states[Connection.CELL_2G] = 'Cell 2G connection';
    states[Connection.CELL_3G] = 'Cell 3G connection';
    states[Connection.CELL_4G] = 'Cell 4G connection';
    states[Connection.NONE] = 'No network connection';

    return states[networkState];
}

```

Inquiry of contact information

```

// Success callback function on device phone number inquiry
function fn_egov_deviceNumber_success(result){

    console.log("DeviceAPIGuide totalFileSystemSize success");
    deviceInfoVO.telno = result;
    $('deviceInfo:eq(3)').html(deviceInfoVO.telno);

    return result.telno;
}

```

Memory information inquiry

```

// Success callback function on device memory inquiry
function fn_egov_totalSpace_success(result){

```

```

console.log("DeviceAPIGuide totalFileSystemSize success");
deviceInfoVO.strgeInfo = result;
$('.deviceInfo:eq(5)').html(deviceInfoVO.strgeInfo);

return result.totalSpace;
}

```

Related Screen and Implementation Manual

Device information inquiry screen



OS: Platform info and version

UUID : Device's ID

PhoneGap Version : PhoneGap Version applied to DeviceInfo Guide Program

Contacts : Device phone number

Connection Type : Device network status info
Total Memory : Device total memory

Send device information to server

Business Logic

- Check the network. If the network is not Wi-Fi, function based on the user approval.
- InterfaceAPI is utilized to send the device information to server.

Related Codes

Register Device information

```
function fn_egov_regist_deviceInfo() {  
  
    useYn = "Y";  
  
    var url = "/dvc/xml/addDeviceInfo.do";  
    var accept_type = "json";  
  
    var params = {  
        uuid : device.uuid,  
        os : device.platform + " " + device.version,  
        telno : String(deviceInfoVO.telno),  
        strgeInfo : deviceInfoVO.strgeInfo,  
        ntwrkDeviceInfo : deviceInfoVO.ntwrkDeviceInfo ,  
        pgVer : device.cordova};  
  
    // send the data  
    egovHyb.post(url, accept_type, params, function(jsondata) {  
        var data = JSON.parse(jsondata);  
  
        if(data.useYn == "OK"){  
            //fn_goDeviceInfoListXml();  
        }else{  
            $("#alert_dialog").click( function() {  
                jAlert('Error occured while sending data.', 'Transmission  
Error', 'c');  
            });  
        }  
  
    });  
  
    console.log("DeviceAPIGuide fn_egov_regist_deviceInfo request Completed");  
  
}
```

Related Screen and Implementation Manual

Action	URL	Controller method	QueryID
Register Device information	/dvc/addDeviceInfo.do	insertDeviceInfo	“deviceaddAPIDAO.insertDeviceInfo”

Information upload: Upload device information to server.

Device information list inquiry

Business Logic

- Check the network. If the network is not Wi-Fi, function based on the user approval.

Related Codes

Device information list inquiry

```
function fn_egov_move_deviceInfoList() {
    var url = "/dvc/xml/deviceInfoList.do";
    var accept_type = "xml";
    var params = { uuid : device.uuid };

    // get the data from server
    window.plugins.EgovInterface.get(url,accept_type, params, function(xmldata) {
        var list_html = "";

        $(xmldata).find("deviceInfoList").each(function(){
            var sn = $(this).find("sn").text();
            var uuid = $(this).find("uuid").text();
            var ntwrkDeviceInfo = $(this).find("ntwrkDeviceInfo").text();
            var os = $(this).find("os").text();

            list_html += "<li><a
href='javascript:fn_egov_request_deviceInfoDetail(" + sn + ")'><h3>UUID : " + uuid + "</h3>";
            list_html += "<p><strong>Network Connection Type : " +
ntwrkDeviceInfo + "</strong></p>";
            list_html += "<p>OS : " + os + "</p></a></li>";

        });

        console.log("DeviceAPIGuide fn_egov_move_deviceInfoList request
Completed");

        var theList = $('#theList');
        theList.html(list_html);
        $.mobile.changePage("#deviceInfoList", "slide", false, false);
        theList.listview("refresh");

    });
}
```

Device details inquiry

// requests detailed information about the device saved in server.

```
function fn_egov_request_deviceInfoDetail(data){
    var params = { sn : String(data) };

    var url = "/dvc/xml/deviceInfo.do";
    var accept_type = "xml";

    // get the data from server
    window.plugins.EgovInterface.get(url,accept_type, params, function(xmldata) {
```

```

deviceInfoVO.sn = $(xmldata).find("sn").text();
deviceInfoVO.uuid = $(xmldata).find("uuid").text();
deviceInfoVO.os = $(xmldata).find("os").text();
deviceInfoVO.telno = $(xmldata).find("telno").text();
deviceInfoVO.strgeInfo = $(xmldata).find("strgeInfo").text();
deviceInfoVO.ntwrkDeviceInfo = $(xmldata).find("ntwrkDeviceInfo").text();
deviceInfoVO.pgVer = $(xmldata).find("pgVer").text();
deviceInfoVO.deviceNm = $(xmldata).find("deviceNm").text();

$('.deviceInfo:eq(6)').html(deviceInfoVO.os);
$('.deviceInfo:eq(7)').html(deviceInfoVO.uuid);
$('.deviceInfo:eq(8)').html(deviceInfoVO.pgVer);
$('.deviceInfo:eq(9)').html(deviceInfoVO.telno);
$('.deviceInfo:eq(10)').html(deviceInfoVO.ntwrkDeviceInfo);
$('.deviceInfo:eq(11)').html(deviceInfoVO.strgeInfo);

$("#btnDelDeviceInfo").attr("href","javascript:fn_egov_confirm_delete_deviceInfo(" +
deviceInfoVO.sn + ");");

});

console.log("DeviceAPIGuide fn_egov_request_deviceInfoDetail request Completed");

$.mobile.changePage("#deviceInfoDetail", "slide", false, false);
}
// displays the list of device information returned from server.
function fn_egov_move_deviceInfoList() {
    var url = "/dvc/xml/deviceInfoList.do";
    var accept_type = "xml";
    var params = {uuid : device.uuid };

    // get the data from server
    window.plugins.EgovInterface.get(url,accept_type, params, function(xmldata) {
        var list_html = "";

        $(xmldata).find("deviceInfoList").each(function(){
            var sn = $(this).find("sn").text();
            var uuid = $(this).find("uuid").text();
            var ntwrkDeviceInfo = $(this).find("ntwrkDeviceInfo").text();
            var os = $(this).find("os").text();

            list_html += "<li><a
href='javascript:fn_egov_request_deviceInfoDetail(" + sn + ")'><h3>UUID : " + uuid + "</h3>";
            list_html += "<p><strong>Network Connection Type : " +
ntwrkDeviceInfo + "</strong></p>";
            list_html += "<p>OS : " + os + "</p></a></li>";

        });

        console.log("DeviceAPIGuide fn_egov_move_deviceInfoList request
Completed");

        var theList = $('#theList');
        theList.html(list_html);

```

```

        $.mobile.changePage("#deviceInfoList", "slide", false, false);
        theList.listview("refresh");

    });
}
    Delete Device information

function fn_egov_confirm_delete_deviceInfo(sn){
    jConfirm('Delete device information??', 'Alert', 'c', function(r){
        if(r == true){
            fn_egov_delete_deviceInfo(sn);
        }else{
        }
    });
}

// requests the deletion of device information saved in server.
function fn_egov_delete_deviceInfo(data){
    var url = "/dvc/xml/withdrawal.do";
    var accept_type = "json";

    var params = {sn : String(data)};
    // send the data
    egovHyb.post(url, accept_type, params, function(jsondata) {
        var data = JSON.parse(jsondata);

        if(data.useYn == "OK"){
            $.mobile.changePage("#deviceInfo", { transition: "slide", reverse:
true });
        }else{
            $("#alert_dialog").click( function() {
                jAlert('Error encountered while deleting data.', 'Deletion
error', 'c');
            });
        }
    });

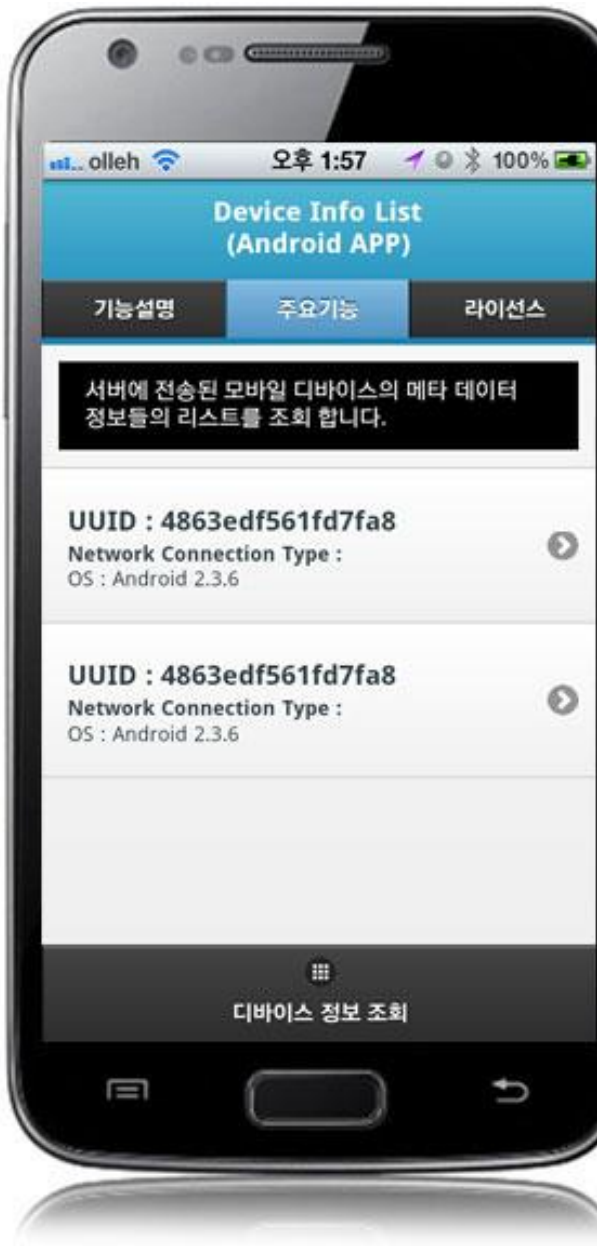
    console.log("DeviceAPIGuide fn_egov_delete_deviceInfo request Completed");
}

```

Related Screen and Implementation Manual

Action	URL	Controller method	QueryID
Device information list inquiry	/dvc/deviceInfoList.do	selectDeviceList	“deviceaddAPIDAO.selectDeviceInfoList”
Device details inquiry	/dvc/deviceInfo.do	selectDeviceInfo	“deviceaddAPIDAO.selectDeviceInfo”
Delete Device information	/dvc/deleteDeviceInfo.do	deleteDeviceInfo	“deviceaddAPIDAO.deleteDeviceInfo”

Device information list inquiry screen



Device details inquiry screen



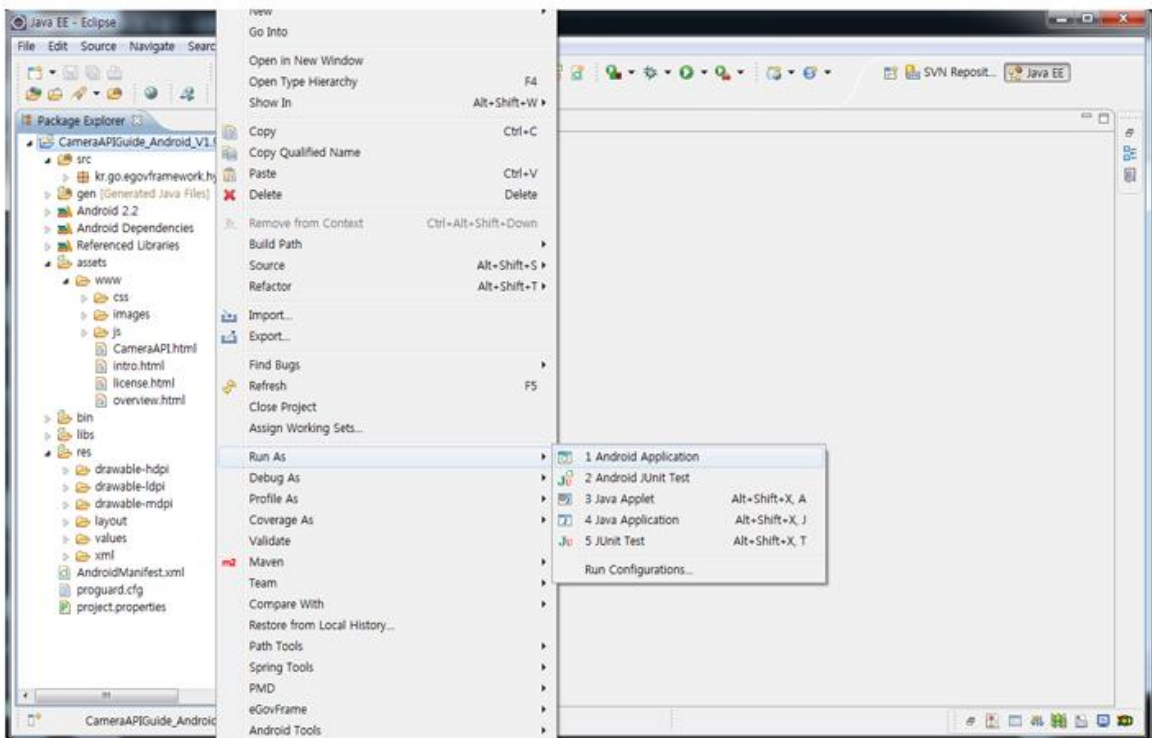
Device information inquiry: requests device information inquiry.
List: calls device information list page.
Delete: deletes device information.

Compiling, debugging, distributing

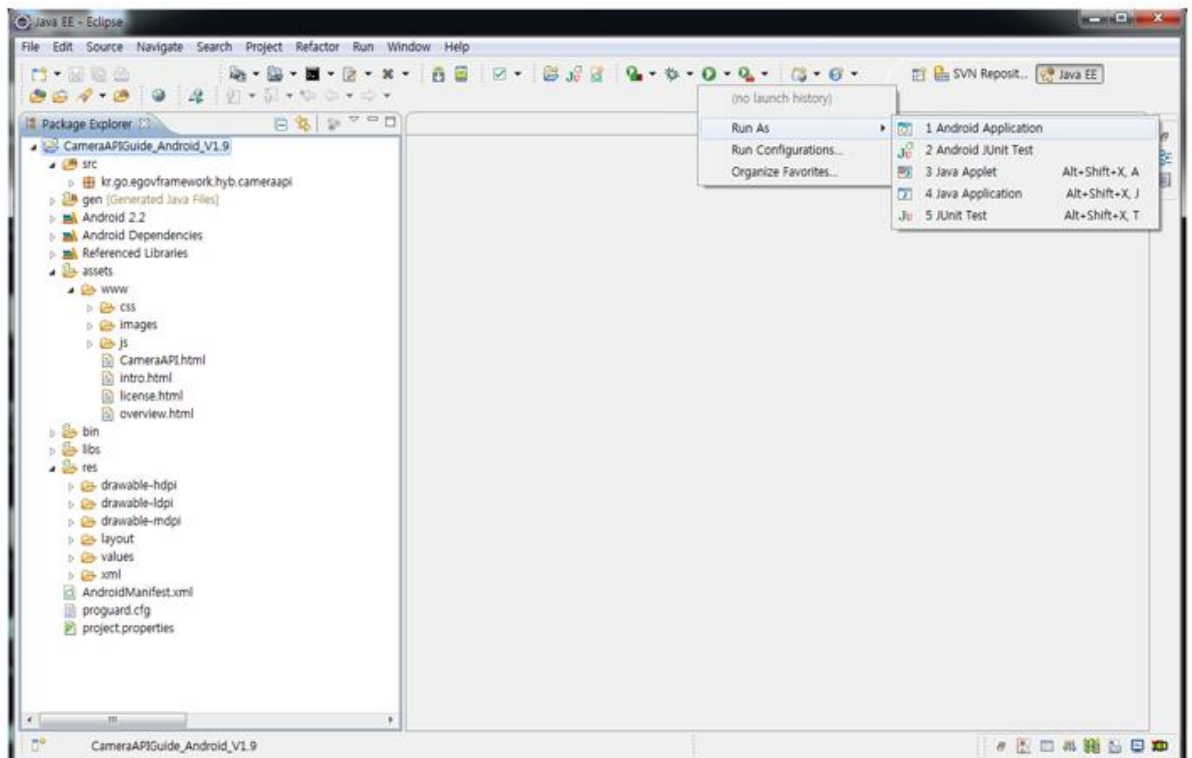
Compiling

How to compile DevicdInfoAPI Device Applicaton

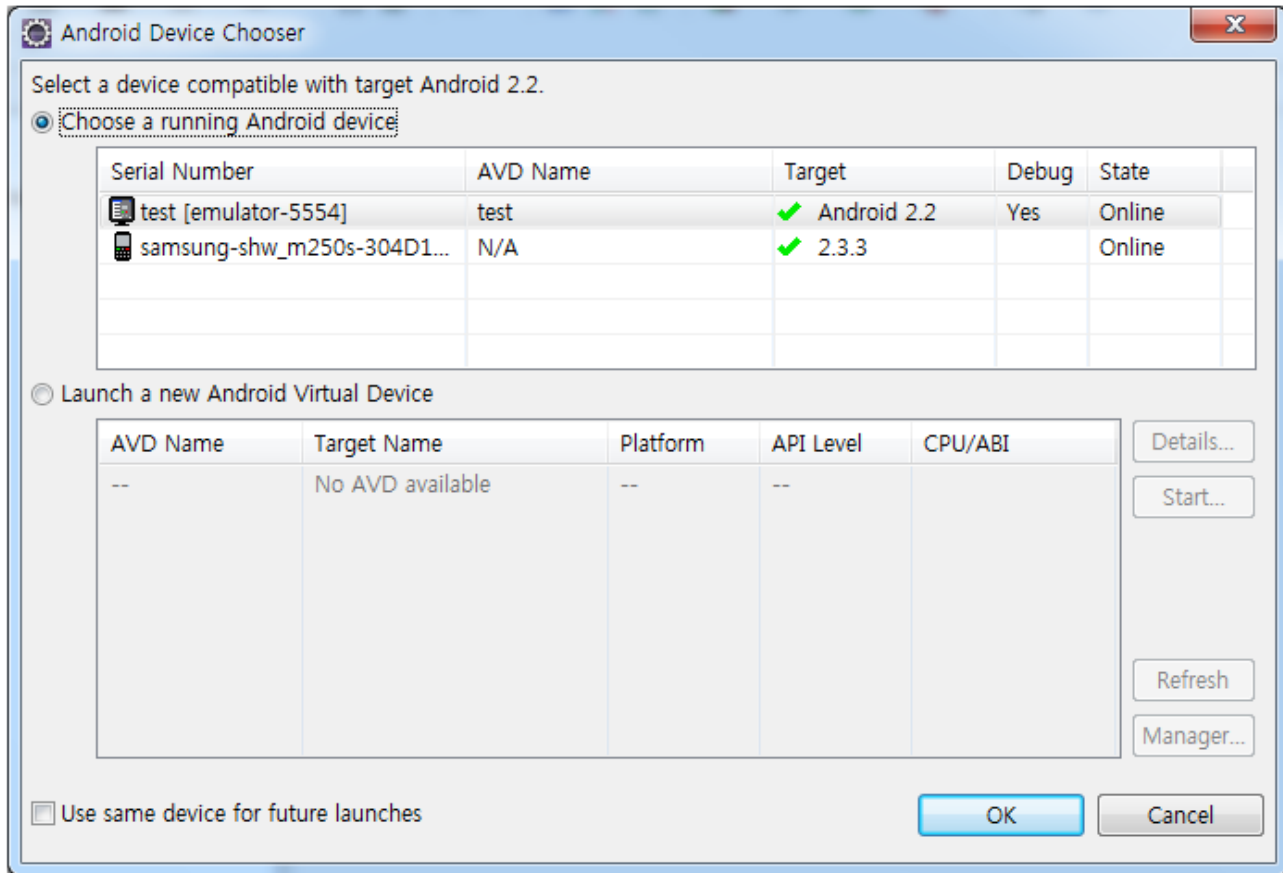
- Choose project > Right-click > Run As > Android Application



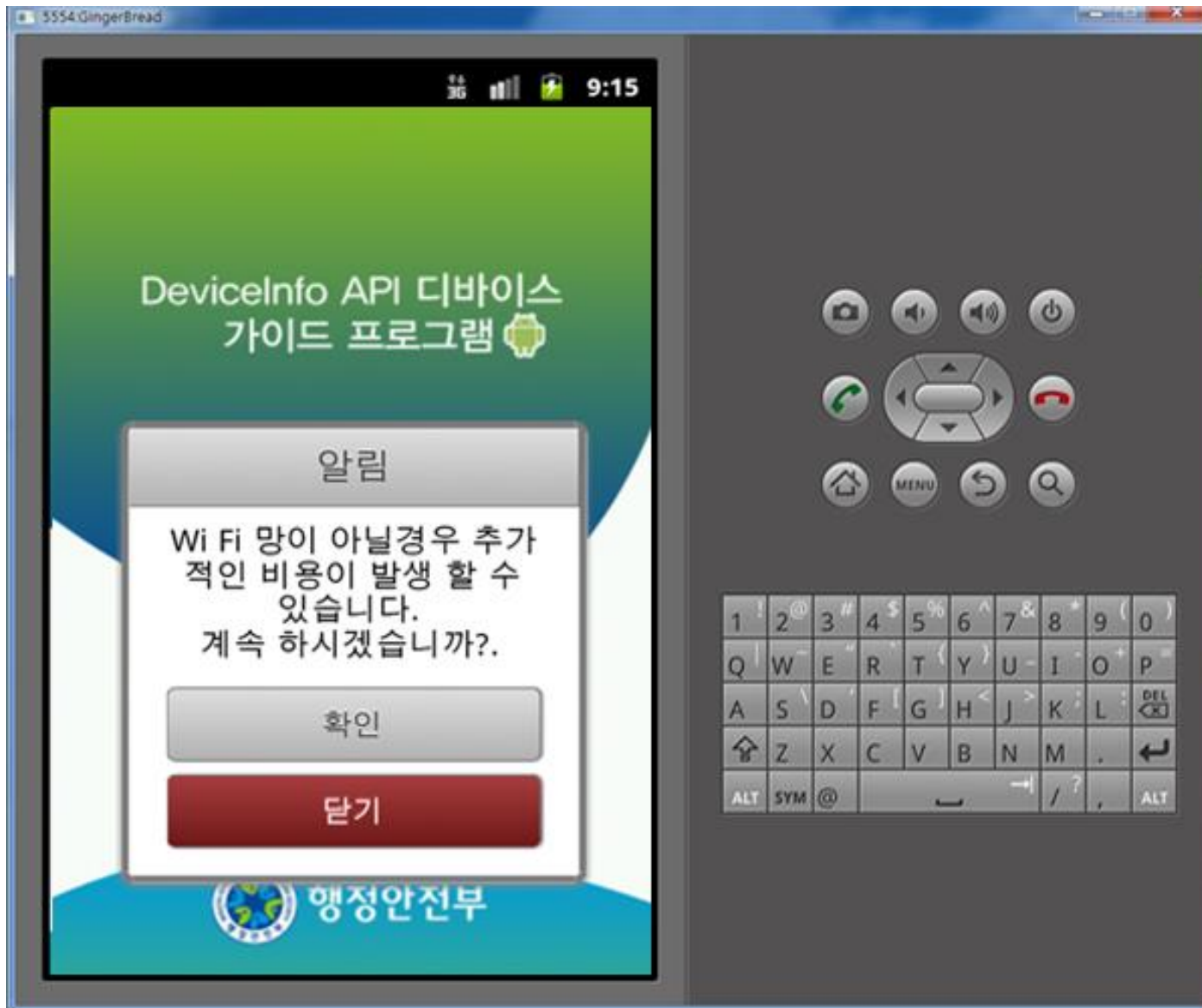
- Top menu > Run As > Android Application



- Select target to run > emulator, Device(debug mode)



- Program display on the emulator

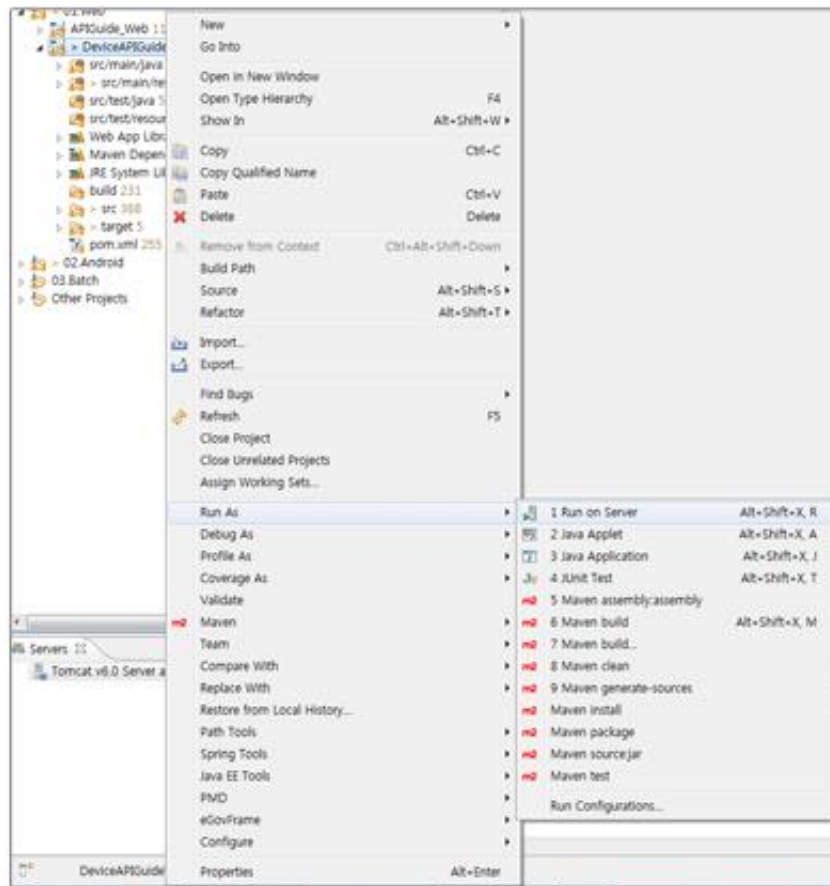


- Program display on the device



How to compile DeviceInfoAPI Server Applicaton

- Right-click on the project and click on Run As>Run On Server in order to run the DeviceInfoAPI 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 'org.springframework.web.serv
2012-09-14 09:15:49,772 INFO [org.springframework.web.servlet.DispatcherServlet] Published WebApplicationContext of servlet 'action' as ServletContext attribute with name [c
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 both Eclipse and Xcode.

Sample for console.log is as follows:

```

function fn_egov_network_check(doCheck)
{
    console.log('DeviceAPIGuide fn_egov_network_check');
}

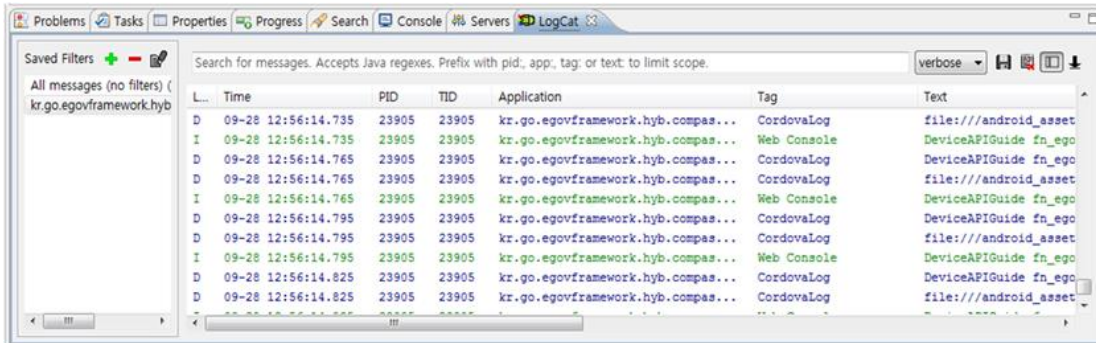
```

```

var networkState = navigator.network.connection.type;
...
}

```

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



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

Debug code	Debug information
DeviceAPIGuide fn_egov_onBatteryStatus Success	Called battery information update success callback function
DeviceAPIGuide fn_egov_onBatteryLow Success	Called low battery callback function
Battery Level Critical ” + info.level + ”%nRecharge Soon!	Called critical battery callback function
DeviceAPIGuide totalFileSystemSize success	Called total device memory inquiry success callback function
DeviceAPIGuide fn_egov_deviceNumber success	Called device phone number inquiry success callback function
DeviceAPIGuide fn_egov_fail” + error	Called custom Plug-In failure callback function
DeviceAPIGuide fn_egov_regist_deviceInfo request Completed	Successfully sent device information to server
DeviceAPIGuide fn_egov_move_deviceInfoList request Completed	Successfully sent device information list to server
DeviceAPIGuide fn_egov_request_deviceInfoDetail request Completed	Successfully sent device details to server
DeviceAPIGuide fn_egov_delete_deviceInfo request Completed	Successfully deleted device information from server

Distribution

Download DeviceInfo Device API Guide: [Click](#)

References

- UX/UI library : jQuery Mobile [Click](#)
- Phonegap 4.3.0 : [Click](#)