

Contacts Device API Guide Program

Outline

Contacts Device API Guide Program supports JavaScript-based control and checking of Contacts related features of mobile smart device.

Also, it serves as a reference and a tool for implementing "back up Contacts information to server" and "restore to back up Contacts information" features.

Feature

This Guide Program provides **Contacts inquiry**, **back up Contacts information**, and **restore to backup Contacts** features.

Preconditions

Category	Description
Local Device Environments	Xcode 6.3.2, PhoneGap 4.3.0
Server-side Developmental Environment	eGov Standard Framework Deveelopment 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	N/A

Restrictions

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);
```

New ID is given when Contacts are created or altered.

- Problem: when restoring Contacts using PhoneGap library, new ID is given to restored Contacts.
- Solution: sync newly restored ID to server.

```
function fn_egov_contactIdUpdate()
{
    if(fn_egov_network_check(false))          // if communication event occurs, determine if user approved 3G
usage
    {
        var params = {contactsList : {contactsList : JSON.stringify(recoveryParam)}};

        $.mobile.showPageLoadingMsg('a');
        // depending on device performance, delay using setTimeout to ensure next code is executed after
        ProgressDialog Show is completely loaded.
        setTimeout(function()
            {
                fn_egov_sendto_server("/ctt/updateContactsiOS.do",params);
            },
            500);
    }
}
```

Using cross domain

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

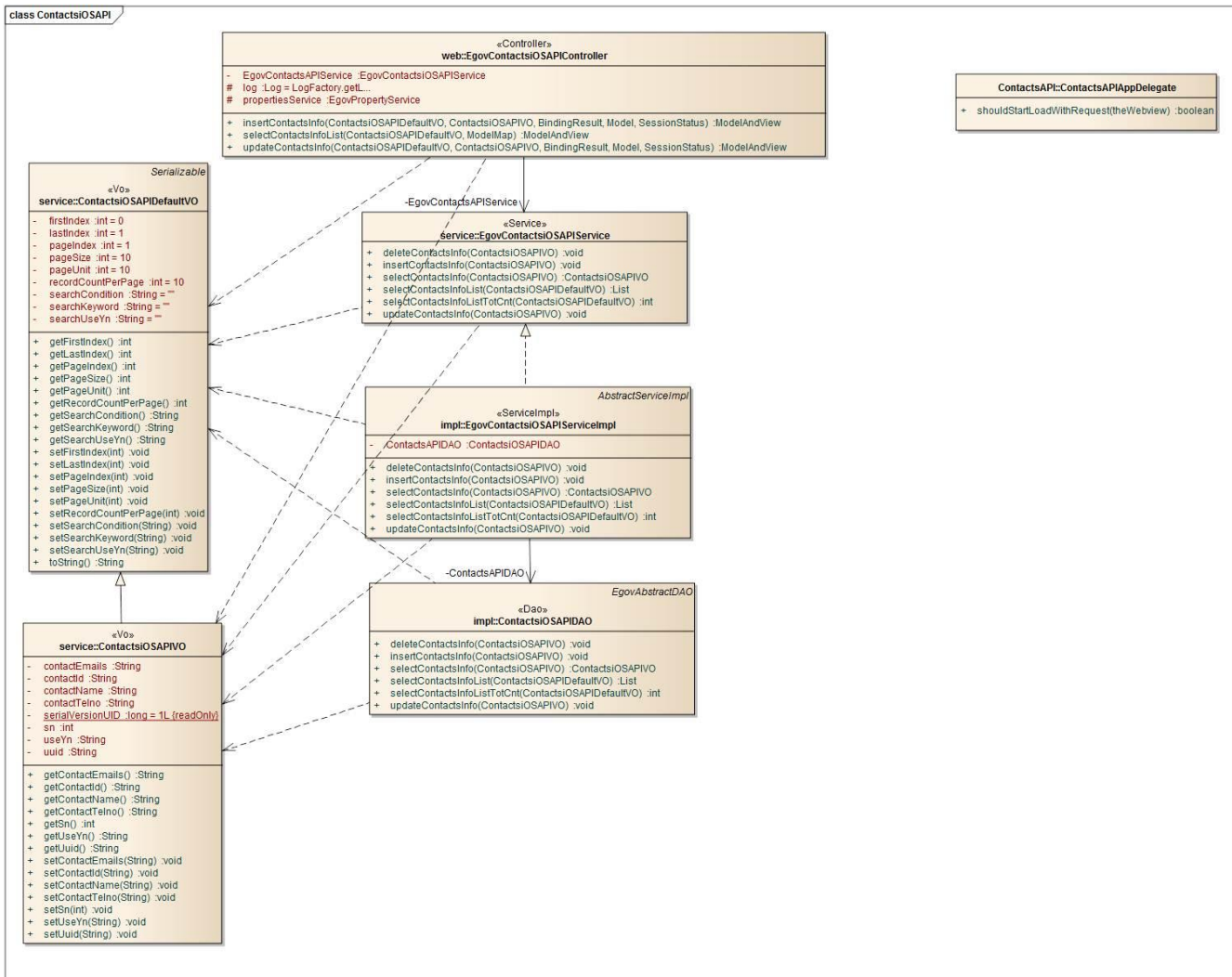
License

N/A

Description

Contacts Device API Guide Program is comprised of **Contacts inquiry, back up Contacts information, and restore to backup Contacts** features.

Related Class Diagram



Device Application

Source

Type	Target Source	Remark
CSS	www/css/egovframework/mb1/hyb/ContactsAPI.css	ContactsAPI Guide Program main Cascading Style Sheets
IMAGE	www/images/egovframework/mb1/hyb/	ContactsAPI Guide Program main Image folder
JS	www/js/egovframework/mb1/hyb/ContactsAPI.js	ContactsAPI Guide Program main JavaScript
HTML	www/ContactsAPI.html	ContactsAPI main page
HTML	www/license.html	ContactsAPI license page
HTML	www/overview.html	ContactsAPI feature description page

APIs Used

contacts.create

- Create a new contact object

```
var myContact = navigator.contacts.create({"displayName": "Test User"});
```

contacts.find

- Finds a contact

```
navigator.contacts.find(contactFields, contactSuccess, contactError, contactFindOptions);
var options = new ContactFindOptions();
options.filter="Bob";
var fields = ["displayName", "name"];
navigator.contacts.find(fields, onSuccess, onError, options);
```

- Saves a contact

```
// create a new contact object
var contact = navigator.contacts.create();
contact.displayName = "Plumber";    // not support iOS
contact.nickname = "Plumber";      //specify both to support all devices

// populate some fields
var name = new ContactName();
name.givenName = "Jane";
name.familyName = "Doe";
contact.name = name;

// save to device
contact.save(onSuccess,onError);
```

- Clones a contact

```
var clone = contact.clone();
clone.name.givenName = "John";
console.log("Original contact name = " + contact.name.givenName);
console.log("Cloned contact name = " + clone.name.givenName);
```

- Removes a contact

```
function onSuccess() {
    alert("Removal Success");
};

function onError(contactError) {
    alert("Error = " + contactError.code);
};

// remove the contact from the device
contact.remove(onSuccess,onError);
```

- contactFields

Properties	Object Structure	Remark
ID	(DOMString)	Global Unique Identifier
displayName	(DOMString)	Display name of a contact
name	(ContactName)	Object that contains the entire components
nickname	(DOMString)	Casual name where contact can be made
phoneNumbers	(ContactField [])	Phone contact field

email	(ContactField [])	E-mail contact field
addresses	(ContactAddress [])	Address field
ims	(ContactField [])	IM Address field
organizations	(ContactOrganization [])	Organization field
birthday	Date	Birth date of the contact (birthdate)
note	(DOMString)	Remark on the contact
photo	(ContactField [])	Photo field
categories	(ContactField [])	Customized array
urls	(ContactField [])	Web page array

- ContactName

Properties	Object Structure	Remark
formatted	(DOMString)	Full name of the contact
familyName	(DOMString)	
givenName	(DOMString)	
middleName	(DOMString)	
honorificPrefix	(DOMString)	Honorific prefix
honorificSuffix	(DOMString)	Honorific suffix

- ContactField

Properties	Object Structure	Remark
type	(DOMString)	Field category arrays
value	(DOMString)	Field value
pref	(Boolean)	Preferred value

- ContactAddress

Properties	Object Structure	Remark
pref	(boolean)	Representative value of ContactAddress
type	(DOMString)	Field types
formatted	(DOMString)	Address information for output
streetAddress	(DOMString)	Street information

locality	(DOMString)	City / County
region	(DOMString)	State / Region
postalCode	(DOMString)	Postal code
country	(DOMString)	Area code

- ContactOrganization

Properties	Object	Structure	Remark
pref	(Boolean)		Representative value of ContactAddress
type	(DOMString)		Definition of field types (e.g. 'work')
name	(DOMString)		Name of the affiliation
department	(DOMString)		Department in the affiliation
title	(DOMString)		Title

- ContactFindOption

Properties	Object	Structure	Remark
filter	(DOMString)		Keyword and search conditions (Default: "")
multiple	(Boolean)		Loading of multiple contact information

- ContactError

Properties	Object	Structure	Remark
ContactError.UNKNOWN_ERROR	(DOMString)		Unknown error
ContactError.INVALID_ARGUMENT_ERROR	(DOMString)		Wrong argument
ContactError.TIMEOUT_ERROR	(DOMString)		Timeout error
ContactError.PENDING_OPERATION_ERROR	(DOMString)		Invalid command
ContactError.IO_ERROR	(DOMString)		I/O error
ContactError.NOT_SUPPORTED_ERROR	(DOMString)		Not supported
ContactError.PERMISSION_DENIED_ERROR	(DOMString)		Permission denied

Server Application

Source

Type	Target	Source	Remark
Controller	egovframework.hyb.ios.ctt.web.EgovContactsiOSAPIController.java		Classes for Contacts information administration.
Service	egovframework.hyb.ios.ctt.service.EgovContactsiOSAPIService.java		Service interface for Contacts information admin

istration.

VO	egovframework.hyb.ios.ctt.service.ContactsiOSAPIDefaultVO.java	VO Class for Contacts administration.
VO	egovframework.hyb.ios.ctt.service.ContactsiOSAPIVO.java	VO Class for Contacts administration.
DAO	egovframework.hyb.ios.ctt.service.impl.ContactsiOSAPIDAO.java	Data processing classes for Contacts administration.
DAO	egovframework.hyb.ios.ctt.service.impl.EgovContactsiOSAPIServiceImpl.java	Class for processing services requested.
Query XML	resources/egovframework/sqlmap/hyb/ios/ctt/EgovContactsiOSAPIGuide_SQL_mysql.xml	Query file for Contacts administration.

Related Tables

Title

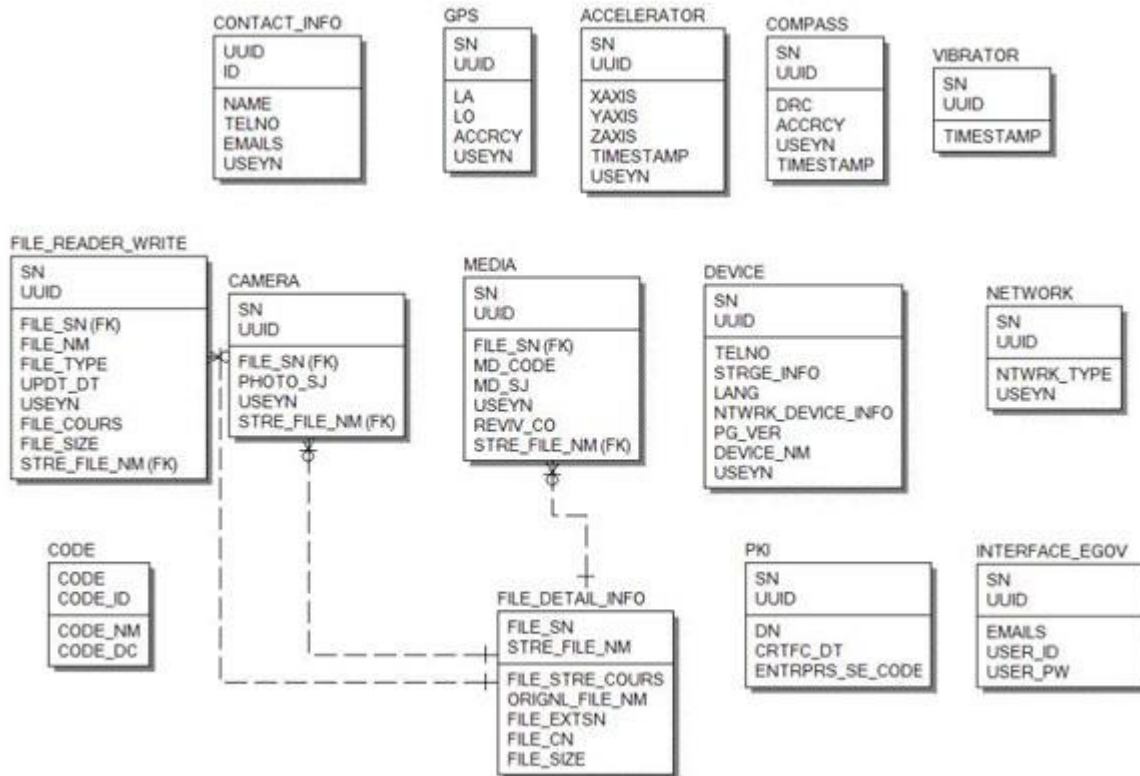
Title	Table	Remark
Contact Information	Contacts	Contact backup information

Tables Breakdown

Contact Information

No.	Column	Title of Column	Type	Length	Null	KEY
1	ID	Serial No.	VARCHAR	20	NotNull	pk
2	UUID	UUID	VARCHAR	50	NotNull	pk
3	NAME	Title	VARCHAR	60	Null	
4	TELNO	Contact Phone	VARCHAR	20	Null	
5	EMAILS	Email	VARCHAR	50	Null	
6	USEYN	Activation	CHAR	1	Null	

ERD



Configuration

Contacts Device properties required for use of GPS-related functions of Contacts Device API Guide Program are as follows:

Device Application

config.xml

```

<featurename="Contacts">
<paramname="ios-package" value="CDVContacts"/>
</feature>
  
```

Server Application

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

```

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

Related Features

Contacts Device API Guide Program is comprised of **Contacts inquiry**, **back up Contacts information**, and **restore to backup Contacts** features.

Contacts inquiry

Business Logic

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

Related Codes

Inquires a Contact

```
function fn_egov_move_localContactList()
{
    // check user approval at initial execution
    if(check3G)
    {
        if(fn_egov_network_check(true))           // determine if user approved 3G usage at initial execution
        {
            check3G = false;
        }
        return;
    }

    $.mobile.showPageLoadingMsg('a');
    var obj = new ContactFindOptions();
    obj.filter = "";
    obj.multiple = true;
    var fields = ["*"];
    navigator.contacts.find(fields, fn_egov_contactListSuccess, fn_egov_fail, obj);
}

    Contact inquiry successful
```

```
function fn_egov_contactListSuccess(contacts)
{
    console.log("DeviceAPIGuide fn_egov_contactListSuccess");

    var theList = $('#listView');
    theList.html("");
    contactsList = contacts;
    nPageCount = 0;
    fn_egov_showList(true);

    $.mobile.hidePageLoadingMsg('a');
    ProcessName = "contactList";
    $.mobile.changePage("#contactList", "slide", false, false);

    theList.listview("refresh");

    setTimeout(loadiScrollList, 1000);
}

    Detailed inquiry of certain Contact
```

```
function fn_egov_move_localContactDetail(index)
{
    var displayName = "";
    var phoneNumber = "";
    var email = "";
    var ContactName = "";

    if(contactsList[index].name.formatted != null)
    {
        ContactName = contactsList[index].name.formatted;
    }
}
```

```

if(contactsList[index].phoneNumbers != null)
{
    for(var j=0; j<contactsList[index].phoneNumbers.length; j++)
    {
        if(contactsList[index].phoneNumbers[j].type == "mobile")
        {
            phoneNumber = contactsList[index].phoneNumbers[j].value;
            break;
        }
    }
}
if(contactsList[index].emails != null)
{
    for(var j=0; j<contactsList[index].emails.length; j++)
    {
        email = contactsList[index].emails[j].value;
        break;
    }
}

document.getElementById('detailIndex').value = index;
document.getElementById('detailName').value = ContactName;
document.getElementById('detailPhone').value = phoneNumber;
document.getElementById('detailEmail').value = email;
ProcessName = "contactDetail";
$.mobile.changePage("#contactDetail", "slide", false, false);
}

Delete Contact

```

```

function fn_egov_contactLocalDelete() {

var index = $("#detailIndex").val();
contactsList[index].remove(fn_egov_removeSuccess,fn_egov_fail);

}</code>

```

Related Screen and Implementation Manual

Function	_URL_ Controller	method	Display (HTML)
Inquire contact list		fn_egov_contactListSuccess	ContactsAPI.html#contactList
Inquire contact details		fn_egov_move_localContactDetail	ContactsAPI.html#contactDetail
Delete Contact		fn_egov_contactLocalDelete	ContactsAPI.html#contactDetail



List: calls Contact details inquiry.
 + 10 more: displays 10 more Contact lists.
 Delete: deletes a Contact.

Back up Contacts information

Business Logic

1. Check the network. If the network is not Wi-Fi, function based on user approval.
2. When backing up Contacts, update Contacts with same ID and adds Contacts with new ID.

Related Codes

Request total number of Contacts backed up in server.

```
function fn_egov_contactBackupCountSuccess(contacts)
{
    console.log("DeviceAPIGuide fn_egov_contactBackupCountSuccess");
    contactsList = contacts;

    var params = { uuid : device.uuid };

    $.mobile.showPageLoadingMsg('a');
```

// depending on device performance, delay using setTimeout to ensure next code is executed after ProgressDialog Show is completely loaded.

```
setTimeout(function()  
    {  
        fn_egov_sendto_server("/ctt/selectBackupCountiOS.do",params);  
    },  
    500);  
}
```

Backs up Contacts information to server.

function fn_egov_contactBackup()

```
{  
    if(contactsList.length == 0)  
    {  
        return;  
    }  
  
    if(fn_egov_network_check(false)) // if communication event occurs, determine if user approved 3G  
usage  
    {  
        $.mobile.showPageLoadingMsg('a');  
        ProcessName = "contactBackup";  
        var param = [];  
        for(var index = 0; index < contactsList.length ; index++)  
        {  
            var displayName = "";  
            var phoneNumber = "";  
            var email = "";  
  
            if(contactsList[index].name.formatted != null)  
            {  
                displayName = contactsList[index].name.formatted;  
            }  
  
            if(contactsList[index].phoneNumbers != null)  
            {  
                for(var j=0; j<contactsList[index].phoneNumbers.length; j++)  
                {  
                    if(contactsList[index].phoneNumbers[j].type == "mobile")  
                    {  
                        phoneNumber = contactsList[index].phoneNumbers[j].value;  
                        break;  
                    }  
                }  
            }  
  
            if(contactsList[index].emails != null)  
            {  
                for(var j=0; j<contactsList[index].emails.length; j++)  
                {  
                    email = contactsList[index].emails[j].value;  
                    break;  
                }  
            }  
  
            param.push({uuid:device.uuid, id:contactsList[index].id, name:encodeURIComponent(displayName),  
telNo:phoneNumber, useYn:'Y', emails:email});  
        }  
  
        var params = { contactsList : {contactsList : JSON.stringify(param)}};  
        // var params = { contactsList : {contactsList : param}};
```

```

// depending on device performance, delay using setTimeout to ensure next code is executed after
ProgressDialog Show is completely loaded.
setTimeout(function()
{
    fn_egov_sendto_server("/ctt/addContactsiOSInfo.do",params);
},
500);
}
}

```

Related Screen and Implementation Manual

Function	_URL_	Controller	method	Display (HTML)
Inquires total number of backed up contacts	C /ctt/selectBackupCountiOS.do	EgovContactsiOSAPIController	selectContactsCount	ContactsAPI.html#contactBackup
Contact backup request	/ctt/addContactsiOSInfo.do	EgovContactsiOSAPIController	addContactsInfoXml	ContactsAPI.html#contactBackup



Back up Contacts: backs up contacts saved in device to server.

Restore to backup Contacts

Business Logic

1. Check the network. If the network is not Wi-Fi, function based on user approval.
2. When restoring Contacts, delete Contacts in device if ID of Contacts in device and backed up Contacts are the same, and restores to backed up Contacts.

Related Codes

Request total number of Contacts backed up in server.

```
function fn_egov_contactBackupCountSuccess(contacts)
{
    console.log("DeviceAPIGuide fn_egov_contactBackupCountSuccess");
    contactsList = contacts;

    var params = {uuid : device.uuid};

    $.mobile.showPageLoadingMsg('a');
    // depending on device performance, delay using setTimeout to ensure next code is executed after ProgressDialog
    Show is completely loaded.
    setTimeout(function()
    {
        fn_egov_sendto_server("/ctt/selectBackupCountiOS.do",params);
    },
    500);
}
```

Restore to backup Contacts

```
function contactRecoveryCallBack(ContactData)
{
    var id = ContactData["id"];
    var name = ContactData["name"];
    var telNo = ContactData["telNo"];
    var email = ContactData["emails"];

    var options = new ContactFindOptions();
    options.filter=String(id);
    var fields = ["id"];
    navigator.contacts.find(fields,
        function(contacts)
        {
            if(contacts.length != 0)
            {
                contacts[0].remove(function(contact){ },fn_egov_fail);
            }

            var contactname = new ContactName();
            contactname.middleName = name;

            var myContact = navigator.contacts.create({ "name":contactname });

            var phoneNumbers = [2];
            phoneNumbers[0] = new ContactField('work', telNo, false);
            phoneNumbers[1] = new ContactField('mobile', telNo, false);
            phoneNumbers[2] = new ContactField('home', telNo, false);
            myContact.phoneNumbers = phoneNumbers;

            var emails = [2];
            emails[0] = new ContactField('work', email, false);
```

```

        emails[1] = new ContactField('home', email, false);
        myContact.emails = emails;

        myContact.save(function(contact)
        {
            recoveryParam.push({uuid:device.uuid, contactId:id,
newId:contact.id});

            if(listCount === recoveryParam.length)
            {
                $.mobile.hidePageLoadingMsg('a');
                confirm(String(listCount) + 'number of contacts
recovered. Synchronizing saved contact information with server. ');
                $.mobile.showPageLoadingMsg('a');
                fn_egov_contactIdUpdate();
            }
        },fn_egov_fail);

    }, function(contactError){ }, options);
}

    Sync restored Contacts to server

function fn_egov_contactIdUpdate()
{
    if(fn_egov_network_check(false))        // if communication event occurs, determine if user approved 3G
usage
    {
        var params = {contactsList : {contactsList : JSON.stringify(recoveryParam)}};

        $.mobile.showPageLoadingMsg('a');
        // depending on device performance, delay using setTimeout to ensure next code is executed after
ProgressDialog Show is completely loaded.
        setTimeout(function()
        {
            fn_egov_sendto_server("/ctt/updateContactsiOS.do",params);
        },
        500);
    }
}

```

Related Screen and Implementation Manual

Function	_URL_	Controller	method	Display (HTML)
Inquires total number of backed up Contacts	/ctt/selectBackupCountiOS.do	EgovContactsiOSAPIController	selectContactsCount	ContactsAPI.html#contactBackup
Restore to backup Contacts	/ctt/contactsiOSInfoList.do	EgovContactsiOSAPIController	selectContactsInfoListXml	ContactsAPI.html#contactRecovery
Sync restored Contacts to server	/ctt/updateContactsiOS.do	EgovContactsiOSAPIController	updateContactsiOS	ContactsAPI.html#contactRecovery



Restore Contacts: restores with Contacts backed up in server.

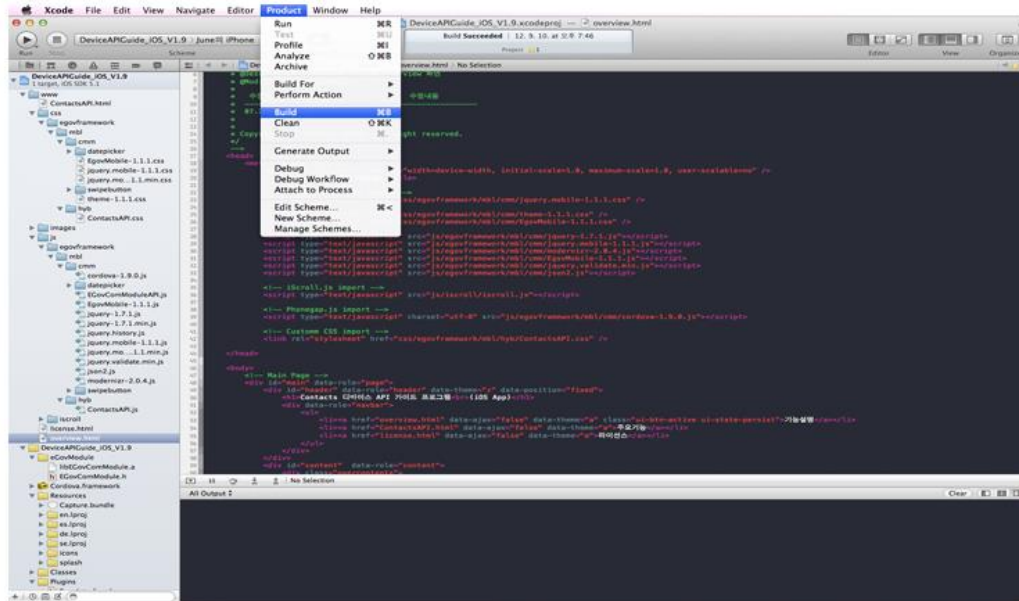
Compiling, debugging, distributing

Compiling

Device Application

1. [ContactsAPI installation link](#)

- Click on "build" after checking ContactsAPI project's installation status.

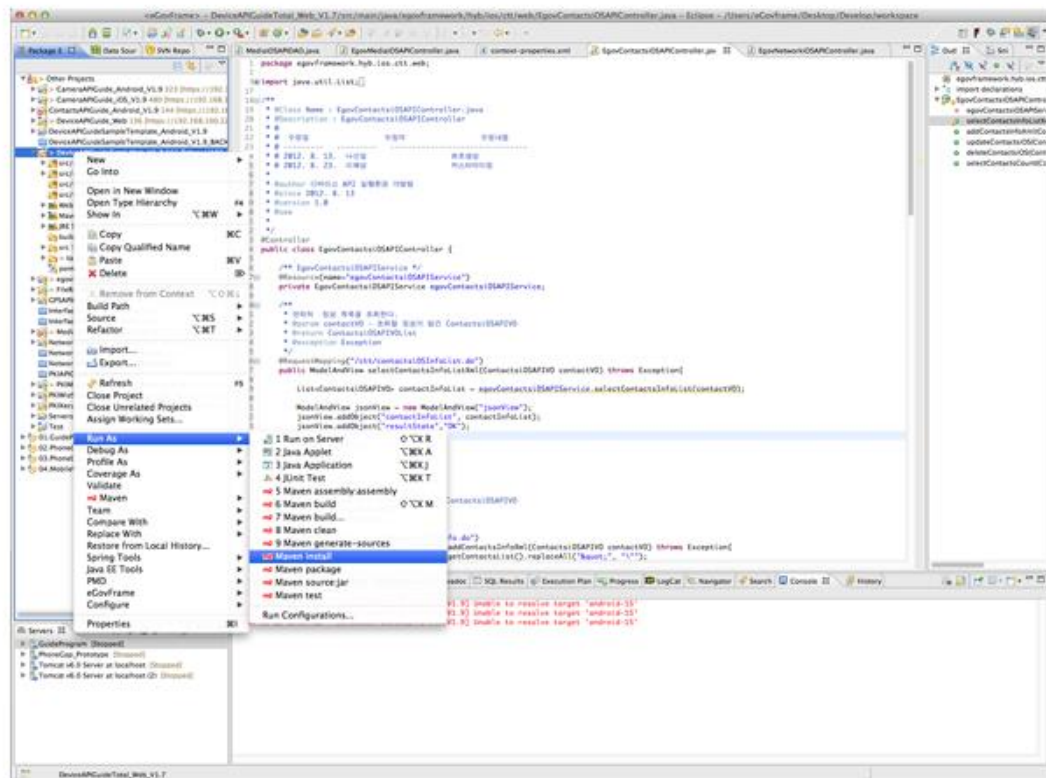


- The following screen will appear upon successful build.

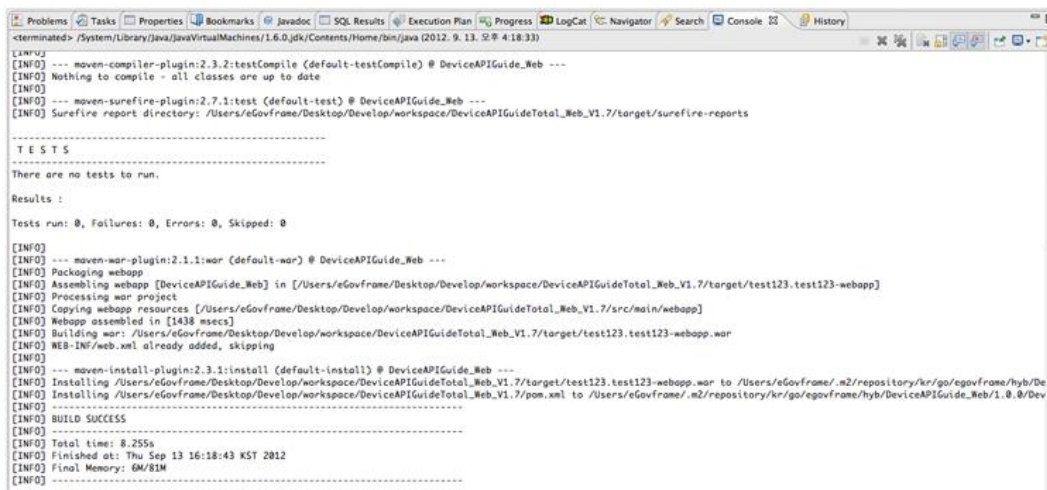


Server Application

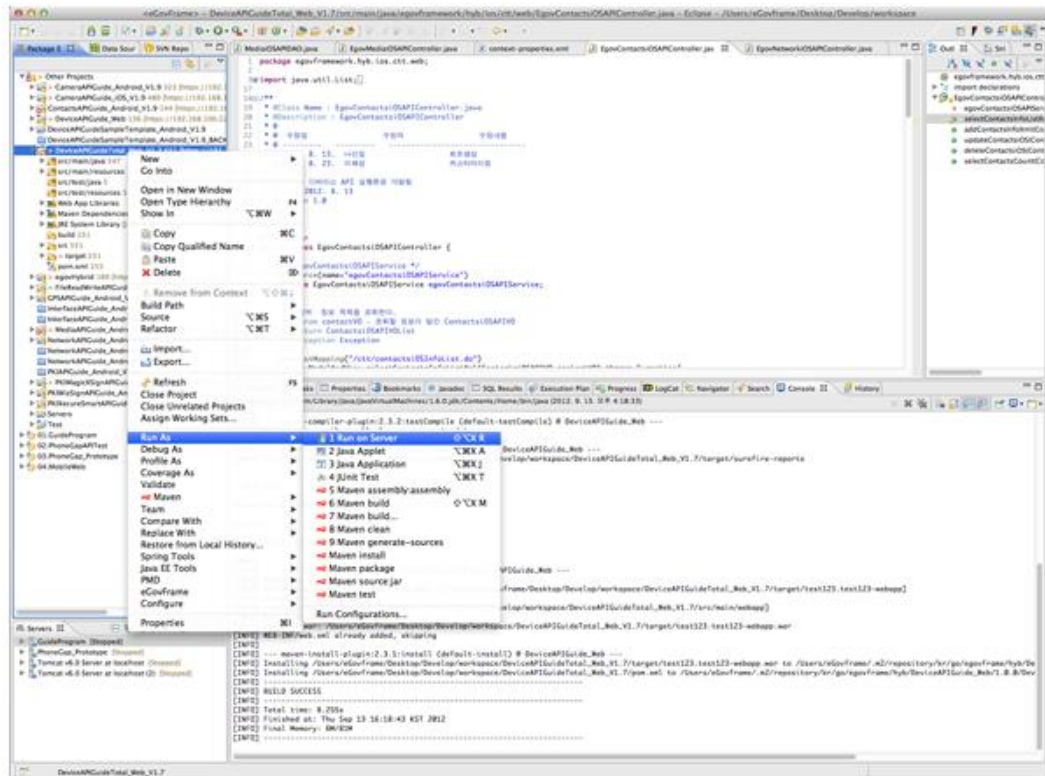
1. [Web Server installation link](#)
2. Right-click on the project>Run as>Maven install to build.



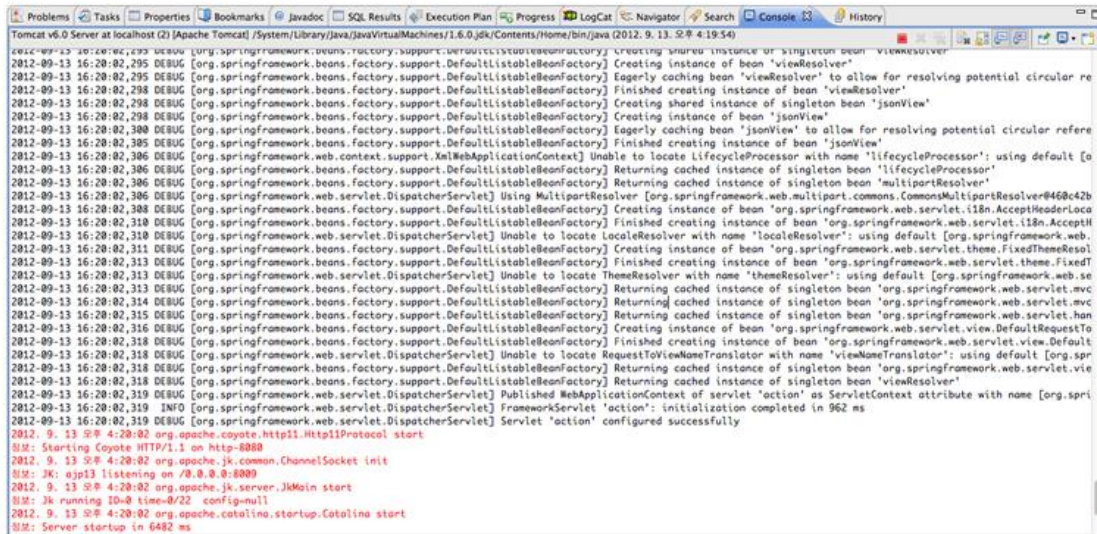
3. Build successful (check console information)



4. Right-click on project>Run as>Run on Server to execute.



5. Check to see if project is successfully executed.



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;
```

```
...
```

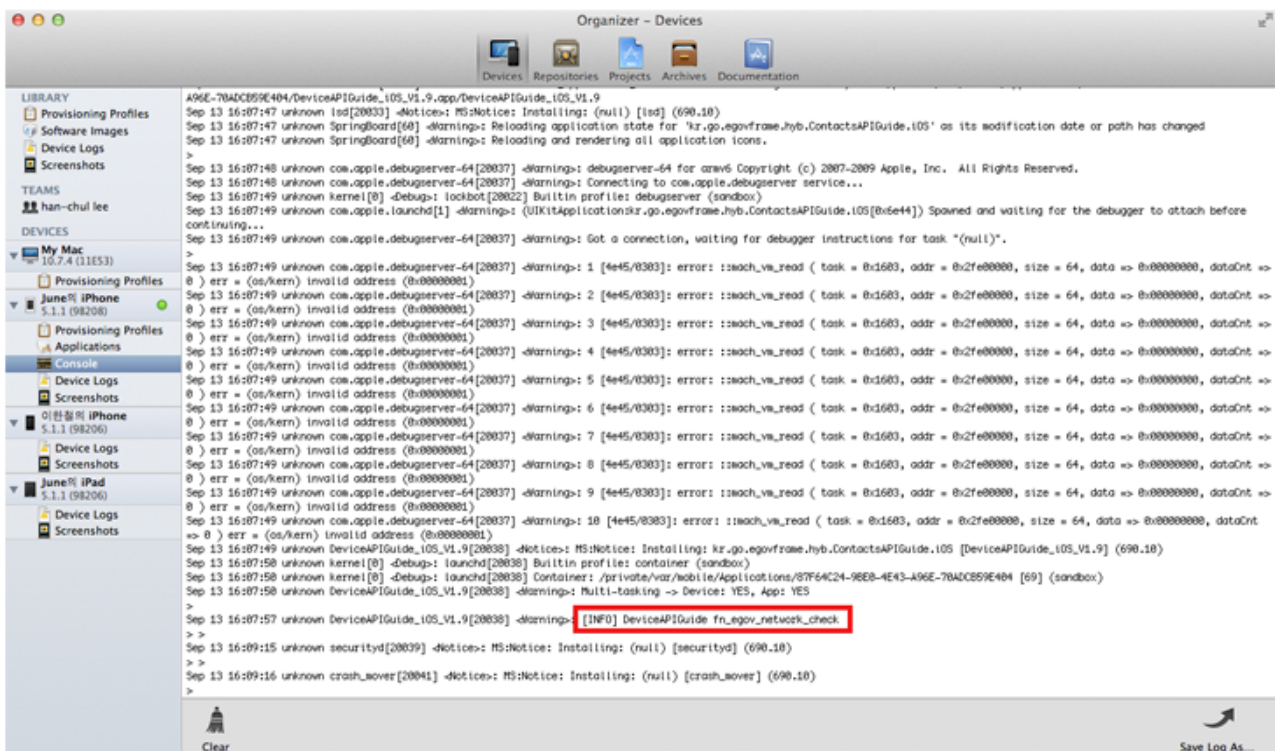
```
}
```

- xCode console window

```

Type "show copying" to see the conditions.
There is absolutely no warranty for GDB. Type "show warranty" for details.
This GDB was configured as "--host=i386-apple-darwin --target=arm-apple-darwin".tty /dev/ttys002
sharedLibrary apply-load-rules all
target remote-mobile /tmp/.XcodeGDBRemote-198-35
Switching to remote-macosx protocol
mem 0x10000 0x3fffffff cache
mem 0x40000000 0xffffffff none
mem 0x80000000 0xbfff none
[Switching to process 7171 thread 0x1c83]
[Switching to process 7171 thread 0x1c83]
2012-09-13 16:07:50.371 DeviceAPIGuide_iOS_V1.9[20038:707] Multi-tasking -> Device: YES, App: YES
2012-09-13 16:07:57.263 DeviceAPIGuide_iOS_V1.9[20038:707] [INFO] DeviceAPIGuide fn_egov_network_check
  
```

- Organizer log window



Log message	Value	Remark
DeviceAPIGuide fn_egov_network_check		Checks network condition
DeviceAPIGuide fn_egov_contactListSuccess		Contacts list in device inquiry successful
DeviceAPIGuide fn_egov_fail + errorCode		Contacts Device API failed due to [error code].
DeviceAPIGuide fn_egov_removeSuccess		Delete Contact successful
DeviceAPIGuide fn_egov_sendto_server Response Completed	Request to server successfully processed	
DeviceAPIGuide fn_egov_sendto_server Response Failed	Internal server process error	

DeviceAPIGuide fn_egov_sendto_server Request Failed

Communications error

DeviceAPIGuide fn_egov_contactBackupCountSuccess

Successfully inquired the number of Contacts in device

DeviceAPIGuide fn_egov_contactRecoveryCountSuccess

Successfully inquired Contacts before restoration

Distribution

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

References

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