

# GPS Device API Guide Program

## Outline

Mobile GPS Device API Guide Program is a guide application for GPS Device API, using the mobile device API framework to be used as a tool and a reference for finding location information and display the location on a map using GPS API when developing hybrid applications.

### Feature

This Guide Program provides **call GPS API information** and **GPS API information DB list** features.

### Preconditions

Category	Description
Local Device Environments	Xcode 6.3.2, PhoneGap 4.3.0
Server-side Developmental Environment	Runtime Environment for eGov Standard Framework 3.5  To use maps provided as an example for GPS API Guide Program, one must apply for the maps' API license key. \\- Daum map: <a href="https://apis.daum.net/register/myapi.daum">https://apis.daum.net/register/myapi.daum</a>
Works in sync with Mash up Open API	All rights to Daum map usage are reserved to applicable institutions. When using Daum map, check ' <b>Daum map Open API service terms of use</b> ' first. (Daum map Open API service terms of use: <a href="http://dna.daum.net/apis/mapagreement">http://dna.daum.net/apis/mapagreement</a> )  Aside from maps used in the example, other map APIs can be connected based on their respective terms of use and standards.
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);
```

Problems may occur if alert() is included in Callback function. (phoneGap)

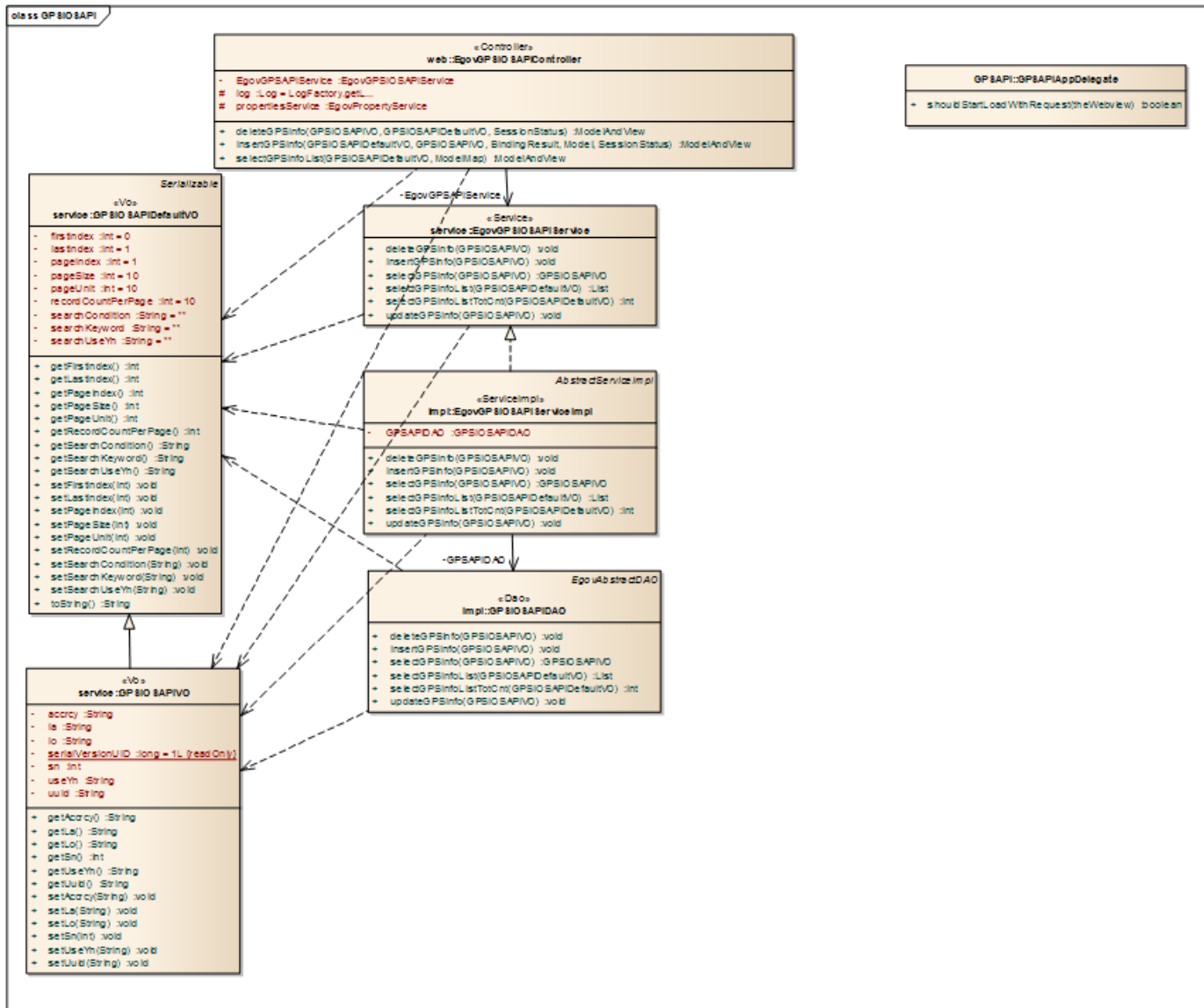
- Problem: error calling alert() message from Callback function saved in PhoneGap.
- Solution: Use asynchronous function or avoid using functions that use thread like alert().

#### **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

## **Description**

## Related Class Diagram



## Device Application

### Source

Type	Target Source	Remark
CSS	www/css/egovframework/mbl/hyb/GPSAPI.css	GPS Guide Program main Cascading Style Sheets
IMAGE	www/images/egovframework/mbl/hyb/	GPSAPI Guide Program main Image folder
JS	www/js/egovframework/mbl/hyb/GPSAPI.js	GPSAPI Guide Program main JavaScript
HTML	www/GPSAPI.html	GPSAPI main page
HTML	www/license.html	GPSAPI license page

HTML [www/overview.html](http://www/overview.html)

GPSAPI feature description page

### APIs Used

`navigator.geolocation.getCurrentPosition`

- PhoneGap API for geolocation information

`navigator.geolocation.getCurrentPosition`

`navigator.geolocation.getCurrentPosition(successCallback, failCallback);`

Option	Description	Remark
--------	-------------	--------

successCallback Function returned upon success

failCallback Returned upon failure

Server Application

### Source

Type	Target Source	Remark
Controller	<code>egovframework.hyb.ios.gps.web.EgovGPSiOSAPIController.java</code>	Classes for GPS information administration.
Service	<code>egovframework.hyb.ios.gps.service.EgovGPSiOSAPIService.java</code>	Service interface for GPS information administration.
VO	<code>egovframework.hyb.ios.gps.service.GPSiOSAPIDefaultVO.java</code>	VO Class for GPS administration.
VO	<code>egovframework.hyb.ios.gps.service.GPSiOSAPIVO.java</code>	VO Class for GPS administration.
DAO	<code>egovframework.hyb.ios.gps.service.impl.GPSiOSAPIDAO.java</code>	Data processing class for GPS administration.
DAO	<code>egovframework.hyb.ios.gps.service.impl.EgovGPSiOSAPIServiceImpl.java</code>	Class for processing services requested.
Query ML	<code>X resources/egovframework/sqlmap/hyb/ios/gps/EgovGPSiOSAPIGui_de_SQL_mysql.xml</code>	Query file for GPS information administration.

### Related Tables

Title	Table	Remark
gps(eGov)	GPS_EGOV	Administration of GPS (eGov) Information

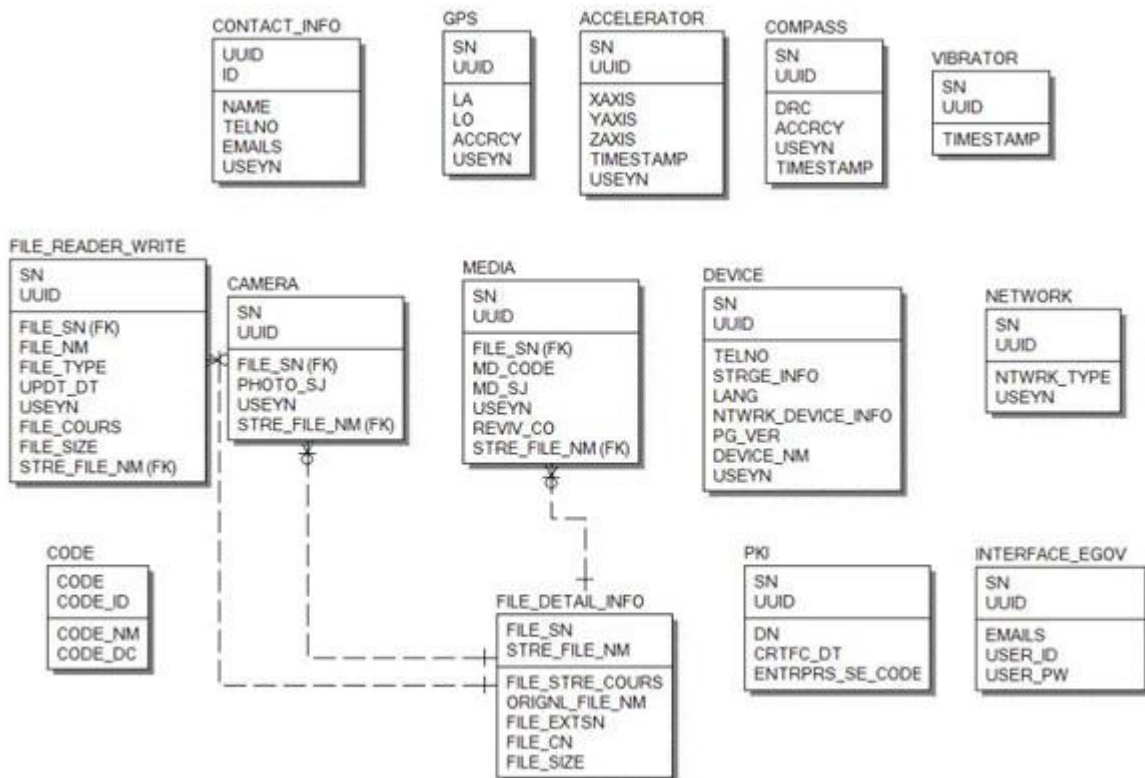
### Table Definition

- GPS

No	Column ID	Title of Column	Type	Length	Null
1	SN	Serial No.	NUMERIC	6	NotNull

2	UUID	UUID	VARCHAR 50	NotNull
3	LA	Latitude value	VARCHAR 40	Null
4	LO	longitude value	VARCHAR 40	Null
5	ACCRCY	Tilt value	VARCHAR 10	Null
6	USEYN	Usage Y/N	CHAR 1	Null

**ERD**



## Configuration Settings

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

**config.xml**

```

<featurename="Geolocation">
<paramname="ios-package" value="CDVLocation"/>
</feature>
<featurename="InterfaceAPI">
<paramname="ios-package" value="EgovInterface"/>
</feature>
    
```

**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/gps/EgovGPSiOSAPIGuide_SQL_[DB  
NAME].xml"/>
```

## Related Features

GPS Device API Guide Program is comprised of **call GPS API information, GPS API information DB list** features.

Call GPS API information

### Business Logic

Calls GPS API information, inquires current location in the map, sends and saves current location to server.

### Related Code

```
// call GPS API information
```

```
navigator.geolocation.getCurrentPosition(onSuccess, onError);
```

```
// display GPS information on the map
```

```
function onSuccess(position) {
```

```
    console.log("GPSAPIGuide navigator.geolocation.getCurrentPosition Success");
```

```
    console.log('Latitude: ' + position.coords.latitude + "\nLongitude: ' + position.coords.longitude);
```

```
    lat = position.coords.latitude;
```

```
    lon = position.coords.longitude;
```

```
    var html = "";
```

```
    html = 'latitude : ' + position.coords.latitude + '<br/>';
```

```
    html += 'longitude : ' + position.coords.longitude + ";
```

```
    $('#latlngInfo').html(html);
```

```
// current location coordinates*
```

```
var latlng = new daum.maps.LatLng(position.coords.latitude, position.coords.longitude);
```

```
// map settings*
```

```
var myOptions = {
```

```
    level: 4,
```

```
    center: latlng,
```

```
    mapTypeId: daum.maps.MapTypeId.ROADMAP
```

```
};
```

```
// generate map*
```

```
var map = new daum.maps.Map(document.getElementById("map"), myOptions);
```

```
map.addControl(new daum.maps.ZoomControl());
```

```

// mark current location with a marker*
var curMarker = new daum.maps.Marker({
    position: latlng
});

curMarker.setMap(map);

var infowindow = new daum.maps.InfoWindow({
    content: '<font
size=2>latitude:'+position.coords.latitude + '<br/>' + 'longitude:'+position.coords.longitude+'</font>'
});

infowindow.open(map, curMarker);

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

// send current location information to server
function fn_egov_go_addGpsInfo()
{
    var params = {uuid : device.uuid,
                  lat: lat,
                  lon: lon,
                  useYn: 'Y'};

    $.mobile.showPageLoadingMsg('a');
    alert('Http Method:POST\nAcceptType:JSON\nSendData:' + JSON.stringify(params));

    EgovInterface.submitAsynchronous(
        [params, "/gps/addGPSInfo.do"],
        function(result) {
            console.log("GPSAPIGuide
fn_egov_go_addGpsInfo 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("GPSAPIGuide
fn_egov_go_addGpsInfo 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**

Function	URL	Controller	method	Display (HTML)
Send GPS information to server	/gps/xml/addGPSInfo.do	EgovGPSiOSAPIController	addGPSInfo	GPSAPI.html#signUp

GPS API information is used for input data.



Click on "Save to DB" to send the information data to the server.



Save to DB: sends GPS data to server.

Back: moves to **GPS Device API Guide Program menu** screen.

GPS API information DB list

### Business Logic

Receives location information list saved in server, performs check, and deletes.

### Related Code

```
// reads list
function fn_egov_go_gpsInfoList()
{
    console.log('fn_egov_go_gpsInfoList()');

    $.mobile.changePage("#gpsInfoList", "slide", false, false);
    $.mobile.showPageLoadingMsg('a');

    EgovInterface.submitAsynchronous(
        ["/gps/gpsInfoList.do"],
        function(result) {
            console.log("GPSAPIGuide
fn_egov_go_gpsInfoList request Completed");

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

            for (var i = 0; i < totcnt; i++) {
                var data = result.gpsInfoList[i];
                list_html += "<li><h3>UUID : " + data.uuid
+ "</h3>";

                list_html += "<p>latitude : " + data.la +
"</p>";

                list_html += "<p>longitude : " + data.lo +
                }
                var theList = $('#theLogList');
                theList.html(list_html);
                theList.listview("refresh");
                $.mobile.hidePageLoadingMsg('a');
                setTimeout(loadiScrollList, 1000);
            },
            function(error) {
                console.log("GPSAPIGuide
fn_egov_go_gpsInfoList 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');
            }
        }
    );
}
```

```

    );

}

// deletes location information list.
function fn_egov_go_deleteGpsInfo()
{
    console.log('fn_egov_go_deleteGpsInfo()');

    jConfirm('SendMethod:RESTful\nSendType:json, post\nParam:\n' + 'url:/gps/deleteGPSInfo.do',
'Alert', 'c', function(r){
        if(r == true){
            $.mobile.showPageLoadingMsg('a');
            EgovInterface.submitAsynchronous(
                ["/gps/deleteGPSInfo.do"],
                function(result) {
                    console.log("GPSAPIGuide

                    console.log("GPS Log List Request

                    var str = '{}';
                    for (myKey in result){
                        str += myKey + ':' +

                    }
                    str += '>';
                    alert('Response

                    //window.history.back();
                    $.mobile.hidePageLoadingMsg('a');
                    location.href = "index.html";

                },
                function(error) {
                    console.log("GPSAPIGuide

                    var str = '{}';
                    for (var myKey in error){
                        str += myKey + ':' +

                    }
                    str += '>';
                    alert('Response

                    $.mobile.hidePageLoadingMsg('a');

                }
            );

        }else{
            return;
        }
    });
}

```

## Related Screen and Implementation Manual

Function	URL	Controller	method	Display (HTML)
Receive GPS location list	/gps/gpsInfoList.do	EgovGPSiOSAPIController	gpsInfoList	GPSAPI.html#gpsInfoList



Click on "Delete from DB" button to delete location information list.

Delete from DB: deletes location information list.

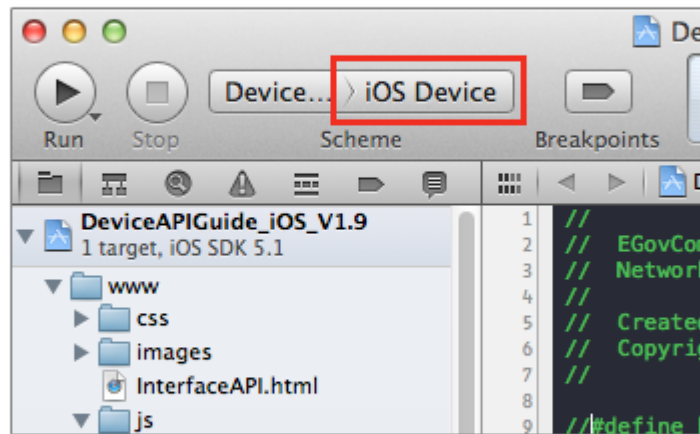
Back: moves to **GPS Device API Guide Program menu** screen.

## Compiling, debugging, distributing

Compiling

**How to compile GPS Device Application**

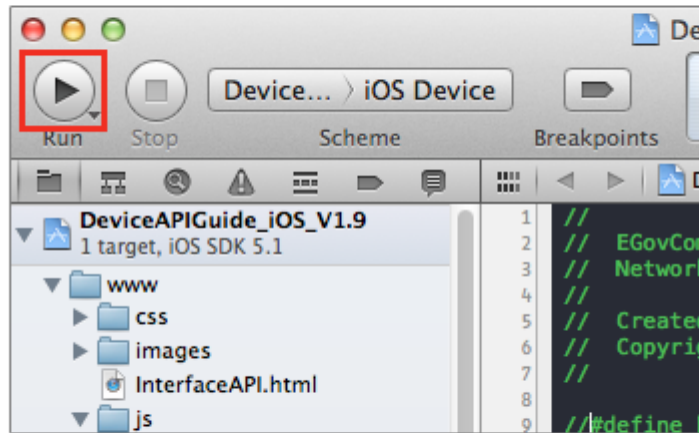
1. To execute on the device or simulator, click on red border area.



2. Select device or simulator.



3. Click on "Execute."



---

4. Check intro and main screen.

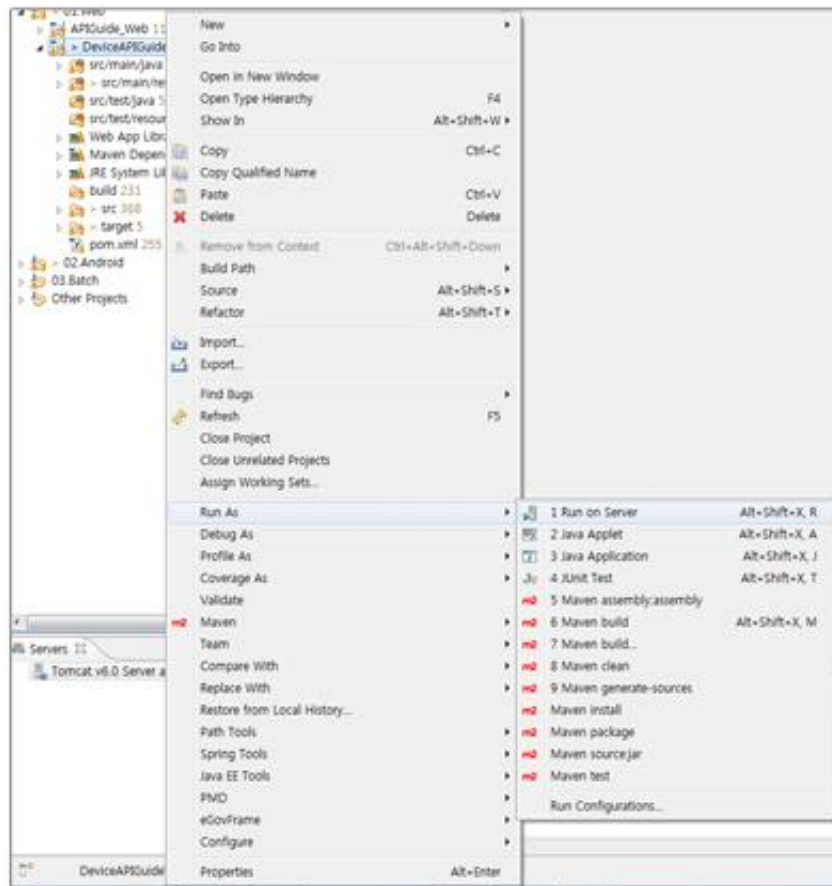




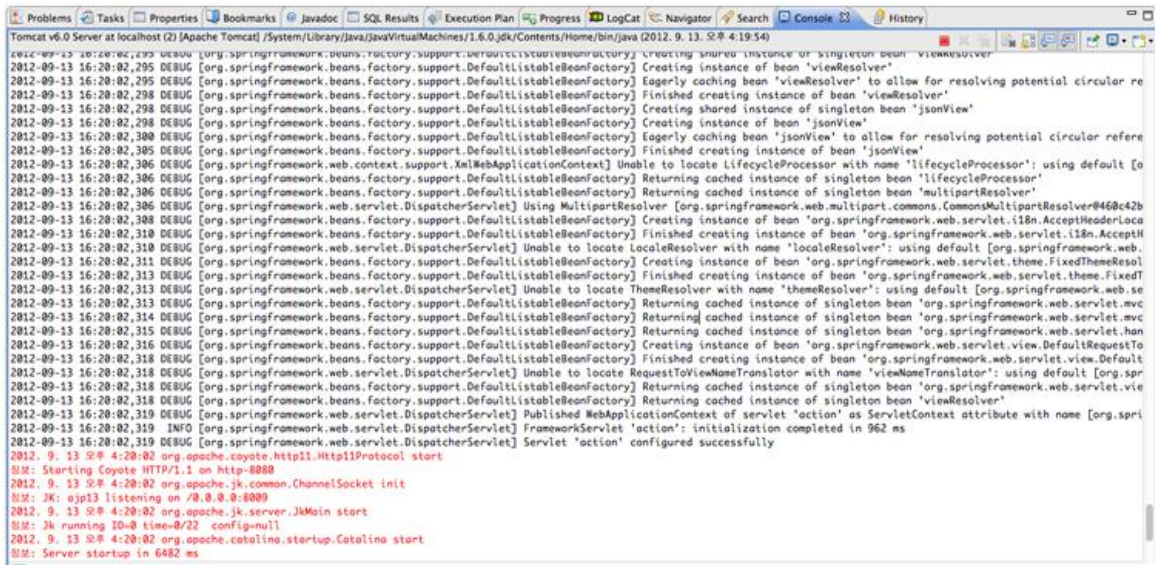
#### **How to compile GPS Server Applicaton**

- Right-click on the project and click on Run As>Run On Server in order to run the GPSAPI 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.



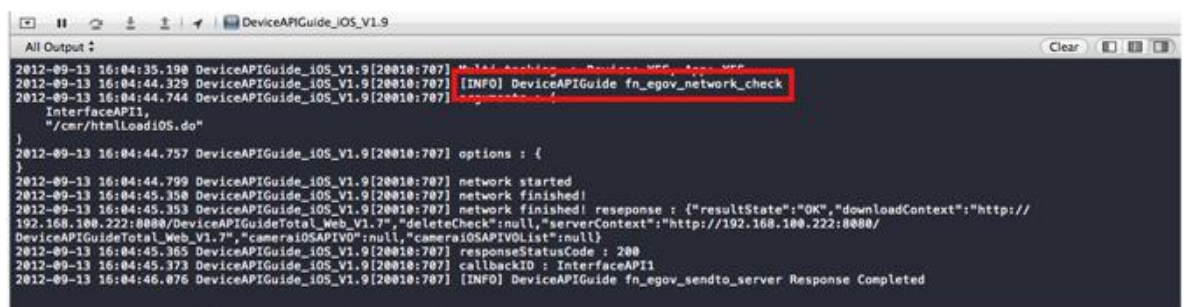
## Debugging

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.

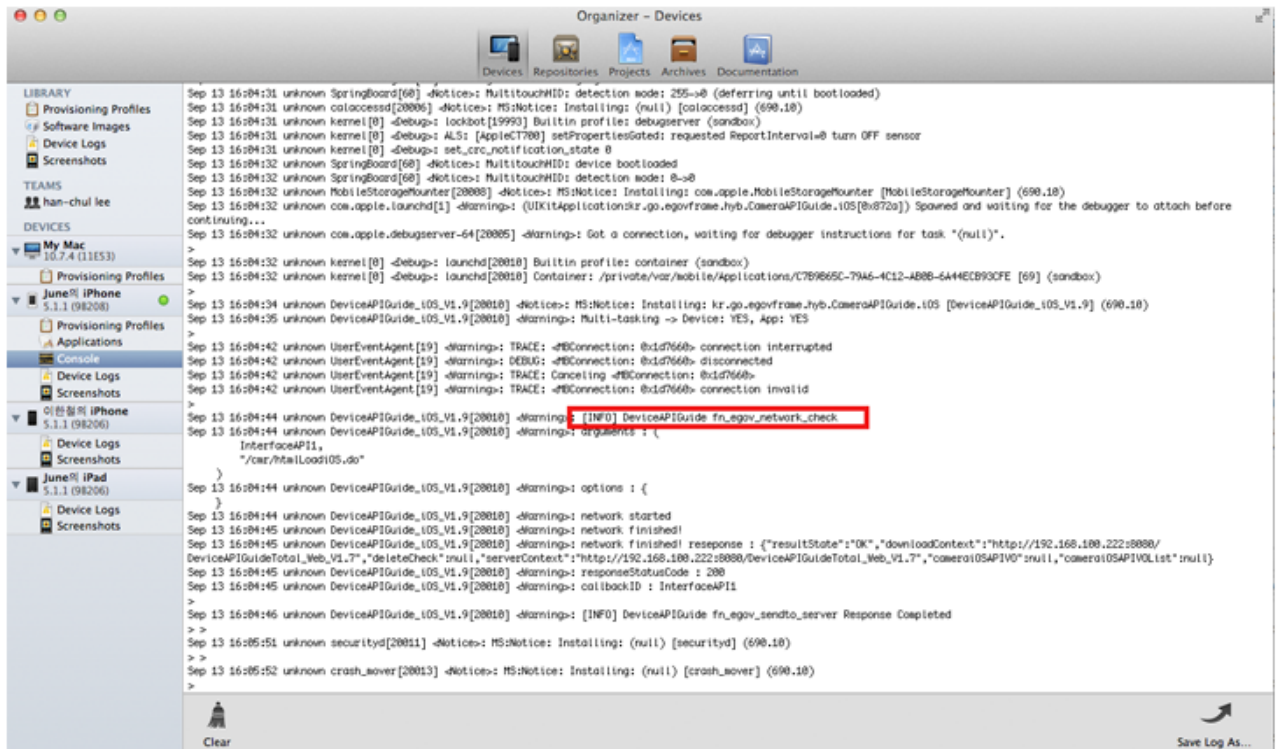
- Example of actual console log

```
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



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

<b>Debug code</b>	<b>Debug information</b>
-------------------	--------------------------

- |  |  |
|--|--|
| GPSAPIGuide navigator.geolocation.getCurrentPosition Success | Successfully read GPS information                  |
| GPSAPIGuide navigator.geolocation.getCurrentPosition Failed  | GPS information reading failed                     |
| GPSAPIGuide fn_egov_go_gpsInfoList request Completed         | Successfully read GPS information list from server |
| GPSAPIGuide fn_egov_go_gpsInfoList request Failed            | Failed to read GPS information list from server    |
| GPSAPIGuide fn_egov_go_addGpsInfo request Completed          | Successfully saved GPS information to server       |
| GPSAPIGuide fn_egov_go_addGpsInfo request Failed             | Failed to save GPS information to server           |
| GPSAPIGuide fn_egov_go_deleteGpsInfo request Completed       | Successfully deleted GPS list from server          |
| GPSAPIGuide fn_egov_go_deleteGpsInfo request Failed          | Failed to delete GPS list from server              |

Distribution

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

## References

- UX/UI library : jQuery Mobile [Click](#)
- Phonegap 4.3.0 : [Click](#)
- Daum Map : <http://dna.daum.net/apis/maps>