Getting Started with iClient for Android

SuperMap iClient for Android is a set of application program interface (API) for the Android 2.0 and above devices, which not only provides the basic interfaces, but also map query, feature plotting and other services. The basic procedures for map development with SuperMap iClient for Android are as follows:

- Development Environment Requirements
- 1. Configuring Eclipse
- 2. Creating Projects
- 3. Adding Dependent Libraries
- 4. Code implementations
- 5. Compiling and Running

Development Environment Requirements

1. JDK

Version 1.6 and above. This example uses JDK 1.6.

Eclipse needs requires Java Runtime Environment (JRE), included in the JDK (Java Development Kit). Android SDK relies on JDK, so you need to prepare JDK firstly.

If you use a non-install JDK, you need to create a new variable JAVA_HOME in the system context variable, set its value as the directory of JDK, and add the value
"%JAVA_HOME%\bin" and "%JAVA_HOME%\lib" in the system context variable "Path".

2. Eclipse

   Version 3.6.2 and above. This example uses eclipse 3.6.2.
   Eclipse is common Java IDE (Integrated Development Environment). We will use Eclipse as Java IDE in the example.

3. Android SDK

   Version 2.0 and above. This example uses Version 2.2.
   Android SDK provides a series of API libraries and programming tools, which are used to construct, test and debug applications.
   If you use a non-install Android SDK, you need to unzip it and then use an environment variable to create a new variable "%ANDROID_SDK_HOME%" whose value is the root directory of Android SDK. You run "SDK Manager.exe" under the root directory to download platforms and packages required by the Android platform.

4. ADT Plugin

   Version 10.0 and above. This example uses Version 10.0.
   ADT (Adroid Development Tools for Eclipse) is a required plugin for Android development based on Eclipse IDE, providing IDE (Integrated Development Environment) for constructing Android applications.
   Open Eclipse, and then click "Help" > "Eclipse Marketplace..." to search and install Android Development Tools for Eclipse.

1. Configuring Eclipse

1. Configuring JRE

   Open Eclipse, then select "Windows" > "Preferences...", then select "Java"> "Installed JREs" on the left side of the menu in the Preferences dialog box. Click the "Add..." button to add a new JRE; click "Edit..." to update the original JRE object.
2. Configuring an Android Plugin

If you use ADT Bundle, then you do not need to configure it anymore because ADT Bundle already includes the configured Eclipse, ADK and Android SDK.

If you use a stand-alone Eclipse, ADK and Android SDK, you need to configure it. The specific method is: Select "Android" on the left side of the menu in the Preferences dialog box, then input the location of Android SDK on the right side of "SDK Location", then select a supported Android platform, and then click "Apply" and "OK".
3. Creating an AVD Emulator

Select "Windows" > "AVD Manager", and then click the "New..." button in the pop-up "Android Virtual Device Manager" to create a AVD emulator, and then input the related parameters such as name, supported platform, size of SD card, resolutions and hardware.
Click "Create AVD".

Select the new "AndroidTest" simulator, and click "Start" and set the display size in the pop-up dialog box (optional), and then click "Launch".
Refer to

- 2. Creating Projects
- 3. Adding Dependent Libraries
- 4. Code implementations
- 5. Compiling and Running

2. Creating Projects
• Select "File">“New”>“Project...” on the Eclipse platform

• Select "Android Project" in the pop-up "New Project" dialog box, and then click "Next >"

• Set the project name "GettingStarted", and then click “Next >"

• Select the supported SDK platform, here select "Android 2.2", and then click "Next >"

• Set the name for the application and package like "com.supermap.sample", and then click "Finish"

Refer to

• 1. Configuring Eclipse

• 3. Adding Dependent Libraries

• 4. Code implementations

• 5. Compiling and Running

3. Adding Dependent Libraries

Select the new project "GettingStarted", and then right-click "Properties", and then click "Java Build Path">"Libraries">"Add External JARs..." in the pop-up dialog box to add the dependent JAR package located at [SuperMap iClient for Android Installation Directory]\lib like "iclient-android-maps-****.jar" (the JAR package needs to be located under the libs folder of the current project), and then click "OK".
Refer to

- 1. Configuring Eclipse
- 2. Creating Projects
- 4. Code implementations
- 5. Compiling and Running

Home > Developer guide > Getting started with iClient > Getting started with iClient for Android > 4. Code implementation

4. Code implementations

1. Add permissions in AndroidManifest.xml
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.INTERNET"/>

2. Add Android support in AndroidManifest.xml

<uses-sdk android:minSdkVersion="3" android:targetSdkVersion="8"/>

3. Add map controls in the layout xml (GettingStarted \ res \ layout \ iclient_android_app.xml)

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
     xmlns:android="http://schemas.android.com/apk/res/android"
     android:id="@+id/container"
     android:layout_width="fill_parent"
     android:layout_height="fill_parent"
     android:orientation="vertical" >
     <com.supermap.android.mapsMapView
         android:id="@+id/mapview"
         android:layout_width="fill_parent"
         android:layout_height="fill_parent"
         android:clickable="true"
         android:enabled="true"/>
</RelativeLayout>

4. Let the created GettingStartedActivity (auto-generated) inherit Activity and import the related classes

package com.supermap.sample;
import com.supermap.android.maps.LayerView;
import com.supermap.android.maps.MapView;
import android.app.Activity;
import android.content.res.Configuration;
import android.os.Bundle;
public class GettingStartedActivity extends Activity {
    // Maps provided by SuperMap iServer use a fixed address for delivering
    private static final String DEFAULT_URL = "http://192.168.120.41:8091/iserver/services/map-china400/rest/maps/China";
protected MapView mapView;
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.iclient_android_app);
}
@Override
public void onConfigurationChanged(Configuration newConfig) {
    super.onConfigurationChanged(newConfig);
}
}

5. Initialize GettingStartedActivity and add the following codes in onCreate:

    //Create a mapview
    mapView = (MapView) this.findViewById(R.id.mapview);
    //Create a map layer and point at map services provided by iServer
    LayerView layerView = new LayerView(this);
    layerView.setURL(DEFAULT_URL);
    //Set map scaling
    mapView.setBuiltInZoomControls(true);
    //Load a map layer
    mapView.addLayer(layerView);

Refer to

• 1. Configuring Eclipse
• 2. Creating Projects
• 3. Adding Dependent Libraries
• 5. Compiling and Running
5. Compiling and Running

1. Compiling: Right-click “GettingStarted” and select "Build Project:.

2. Running: After the compiling is completed, right-click “GettingStarted” and then select “Run as”->”1 Android Application”.

3. After the Android emulator starts, click “GettingStarted”, then you can see a figure as shown below:

![Image](image_url)

Refer to

- 1. Configuring Eclipse
- 2. Creating Projects
• 3. Adding Dependent Libraries

• 4. Code implementations