ANPS API

Version	Last modified	Author	Notes
1.0.0 beta	2011-8-31	Jacky	None

1 Android API:

1.1 Introduce

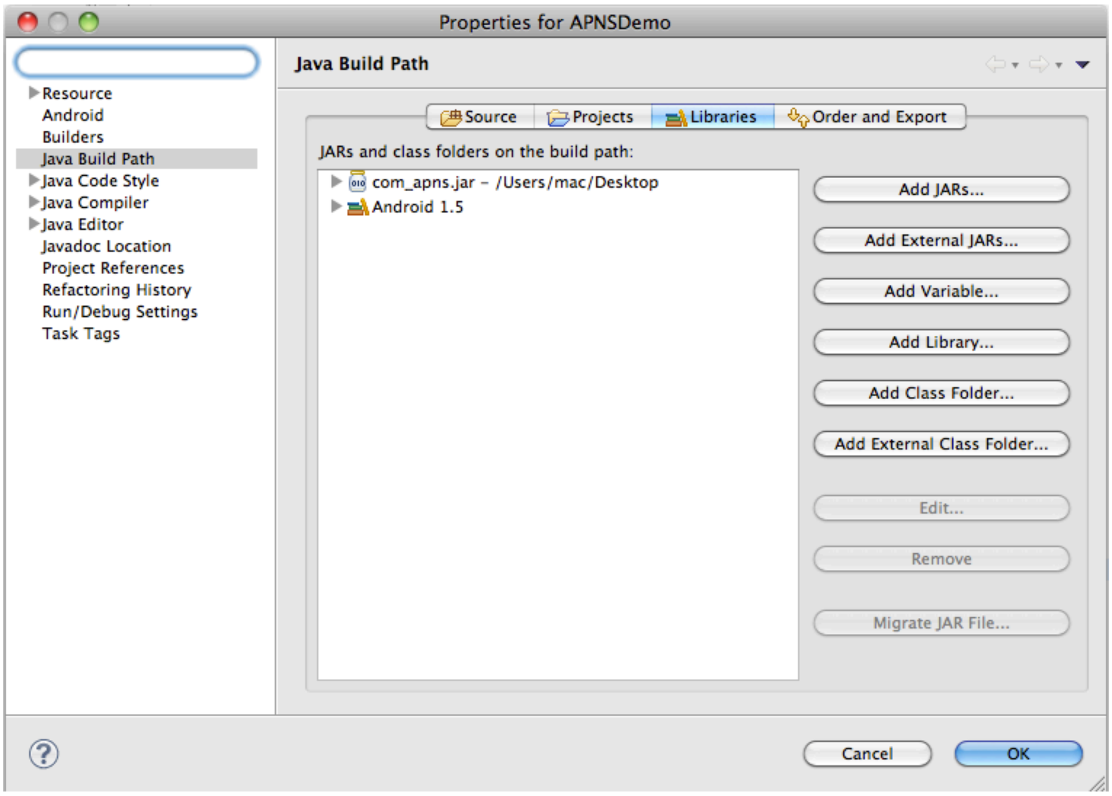
Android Push Notification Service (APNS) is a easy to use libaray for pushing the data to android devices, all you need is to download the libaray and start the service in your app. Once the APNS service is start on device, APNS-Cloud will keep a permanent TCP connection with them, when you need push something to them, send a http rest request to APNS-Cloud, and APNS-Cloud will send the data to the device.

1.2 How to use

1） **Download libaray: apns_x.x.x.jar** (x:version)

2） **Add apns_x.x.x.jar to project**.

Right click the property of project,select “Java Build Path”, -> “Add External JARs”, select **apns_x.x.x.jar**.



Properties for APNSDemo

Java Build Path

JARs and class folders on the build path:

com_apns.jar – /Users/mac/Desktop
Android 1.5

3）**receive push notification**.

Once the service receive a message from server, it will send a broadcast, Use BroadcastReceiver to handle the broadcast.

```
package com.apns.demo;
import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import com.apns.APNService;

public class MyBroadcastReceiver extends BroadcastReceiver {
      @Override
      public void onReceive(Context context, Intent intent) {
           if (intent.getAction().equals(APNService.ON_NOTIFICATION))
           {
                  String str = intent.getStringExtra("data");
                  //todo,
            }
      }
}
```

4）Start the service

Start the service by sending a Intent, which contains the channel name and dev unique id.

```
 Intent intent = new Intent(APNService.START);
intent.putExtra("ch", chanel);
intent.putExtra("devId", devId);
startService(intent);
```

5）Config AndroidManifest.xml

```
<application android:icon="@drawable/icon"
     ...
   <service android:name="com.apns.APNService"
                                    android:label="APNS">
      <intent-filter>
         <action android:name="com.apns.APNService.START" />
         <action android:name="com.apns.APNService.STOP" />
         <category android:name="android.intent.category.DEFAULT"/>
      </intent-filter>
   </service>
   <receiver android:name="MyBroadcastReceiver">
      <intent-filter>
          <action android:name="com.apnsd.APNService.NOTIFICATION" />
      </intent-filter>
   </receiver>
     ...
</application>
<uses-permission
      android:name="android.permission.INTERNET" />
<uses-permission
      android:name="android.permission.ACCESS_NETWORK_STATE" />
```

2 PUSH API:

Send the message to device by using REST interface.

http://www.push-notification.org/handlers/apns_v1.php?ch=YourChannelId&devId=YourDevId&msg =”hello world”&random=0123&hash=HashCode

param:

ch: Channel Id

devId: a unique Id which identify the receiver.

msg: message be send

random: a random number

hash: MD5(ch + g + msg + random + privateKey)

return:

<response result=”0” msg=”sended” />

result 值:

-1: can’t connect to the cloud

0: send seccessfully

1: access failed

2: permission failed

```
 3: device not onling
12: chanel expired
13: hash code not match
14: illege para
15: unkown err
```

3 Attach

3.1 Register url

http://www.push-notification.org

3.2 Demo source code

DemoActivity.Java

```
package com.apns.demo;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.telephony.TelephonyManager;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import com.apns.APNService;


public class DemoActivity extends Activity {
	private Button startButton;
	private Button stopButton;

	@Override
	public void onCreate(Bundle savedInstanceState) {
		super.onCreate(savedInstanceState);
		setContentView(R.layout.main);
		startButton = ((Button) findViewById(R.id.start));
		stopButton = ((Button) findViewById(R.id.stop));
		startButton.setOnClickListener(new OnClickListener() {
			@Override
			public void onClick(View v) {
				String devId = "0";
				// get device id
				TelephonyManager tel = (TelephonyManager)
					getSystemService(Context.TELEPHONY_SERVICE);
				if (tel != null) {
					devId = tel.getDeviceId();
				}
				startAPNS("test",devId);
```

```
                            startButton.setEnabled(false);
                            stopButton.setEnabled(true);
                     }
              });


              stopButton.setOnClickListener(new OnClickListener() {
                     @Override
                     public void onClick(View v) {
                             stopAPNS();
                            startButton.setEnabled(true);
                            stopButton.setEnabled(false);
                     }
              });
       }

       private void startAPNS(String chanel, String devId) {
              Intent intent = new Intent(APNService.START);
              intent.putExtra("ch", chanel);
              intent.putExtra("devId", devId);
              startService(intent);
       }


       private void stopAPNS() {
              Intent intent = new Intent(APNService.STOP);
              stopService(intent);
       }
}
```

MyBroadcastReceiver.class

```
package com.apns.demo;

import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import com.apns.APNService;

public class MyBroadcastReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
       if (intent.getAction().equals(APNService.ON_NOTIFICATION)) {
              String str = intent.getStringExtra("data");
              showNotification(context,str);
       }
    }

     /**
```

```
      *
      *
      * @param text
      */
    private void showNotification(Context context, String text) {
        final NotificationManager manager = (NotificationManager) context
                             .getSystemService(Context.NOTIFICATION_SERVICE);
        Notification notification = new Notification(R.drawable.icon, "APNS",
                             System.currentTimeMillis());
        PendingIntent intent = PendingIntent.getActivity(context, 0, new Intent(
                             APNService.ON_NOTIFICATION), 0);
        notification.setLatestEventInfo(context,
                              "receive new message", text, intent);
        notification.defaults = android.app.Notification.DEFAULT_VIBRATE;
       manager.notify(0, notification);
      }
  }
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
       package="com.apns.demo" android:versionCode="1" android:versionName="1.0">
       <application android:icon="@drawable/icon" android:label="@string/app_name"
              android:debuggable="true">
              <activity android:name=".DemoActivity"
                     android:label="@string/app_name"
                     android:theme="@android:style/Theme.Light.NoTitleBar"
                     android:launchMode="singleTop">
                     <intent-filter>
                            <action android:name="android.intent.action.MAIN" />
                            <category
                            android:name="android.intent.category.LAUNCHER" />
                     </intent-filter>
              </activity>
              <service android:name="com.apns.APNService" android:label="APNS
                     Service Demo">
                     <intent-filter>
                            <action android:name="com.apns.APNService.START" />
                            <action android:name="com.apns.APNService.STOP" />
                            <category
                              android:name="android.intent.category.DEFAULT" />
                     </intent-filter>
              </service>
              <receiver android:name="MyBroadcastReceiver">
                     <intent-filter>
                            <action
                            android:name="com.apnsd.APNService.NOTIFICATION" />
                     </intent-filter>
              </receiver>
       </application>
       <uses-permission android:name="android.permission.INTERNET" />
       <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

```xml
        <uses-permission android:name="android.permission.READ_PHONE_STATE" />
        <uses-permission android:name="android.permission.VIBRATE" />
        <uses-sdk android:minSdkVersion="3" />
</manifest>
```