[**Android调用WebService**](http://www.cnblogs.com/gzggyy/archive/2011/06/21/2086140.html)

WebService是一种基于SOAP协议的远程调用标准，通过webservice可以将不同操作系统平台、不同语言、不同技术整合到一块。在Android SDK中并没有提供调用WebService的库，因此，需要使用第三方的SDK来调用WebService。PC版本的WEbservice客户端库非常丰富，例如Axis2，CXF等，但这些开发包对于Android系统过于庞大，也未必很容易移植到Android系统中。因此，这些开发包并不是在我们的考虑范围内。适合手机的WebService客户端的SDK有一些，比较常用的有Ksoap2，可以从http://code.google.com/p/ksoap2-android/downloads/list进行下载；将下载的ksoap2-android-assembly-2.4-jar-with-dependencies.jar包复制到Eclipse工程的lib目录中，当然也可以放在其他的目录里。同时在Eclipse工程中引用这个jar包。

具体调用调用webservice的方法为：

（1） 指定webservice的命名空间和调用的方法名，如：

SoapObject request =new SoapObject(http://service,”getName”);

SoapObject类的第一个参数表示WebService的命名空间，可以从WSDL文档中找到WebService的命名空间。第二个参数表示要调用的WebService方法名。

（2） 设置调用方法的参数值，如果没有参数，可以省略，设置方法的参数值的代码如下：

Request.addProperty(“param1”,”value”);  
Request.addProperty(“param2”,”value”);

要注意的是，addProperty方法的第1个参数虽然表示调用方法的参数名，但该参数值并不一定与服务端的WebService类中的方法参数名一致，只要设置参数的顺序一致即可。

（3） 生成调用Webservice方法的SOAP请求信息。该信息由SoapSerializationEnvelope对象描述，代码为：

SoapSerializationEnvelope envelope=new   
SoapSerializationEnvelope(SoapEnvelope.VER11);  
Envelope.bodyOut = request;

创建SoapSerializationEnvelope对象时需要通过SoapSerializationEnvelope类的构造方法设置SOAP协议的版本号。该版本号需要根据服务端WebService的版本号设置。在创建SoapSerializationEnvelope对象后，不要忘了设置SOAPSoapSerializationEnvelope类的bodyOut属性，该属性的值就是在第一步创建的SoapObject对象。

（4） 创建HttpTransportsSE对象。通过HttpTransportsSE类的构造方法可以指定WebService的WSDL文档的URL：

HttpTransportSE ht=new HttpTransportSE(“http://192.168.18.17:80  
/axis2/service/SearchNewsService?wsdl”);

(5)使用call方法调用WebService方法，代码：

ht.call（null，envelope）；

Call方法的第一个参数一般为null，第2个参数就是在第3步创建的SoapSerializationEnvelope对象。

（6）使用getResponse方法获得WebService方法的返回结果，代码：

SoapObject soapObject =( SoapObject) envelope.getResponse();

以下为简单的实现一个天气查看功能的例子：

[复制代码](javascript:void(0);)

publicclass WebService extends Activity {   
private static final String NAMESPACE ="http://WebXml.com.cn/";   
// WebService地址   
private static String URL ="http://www.webxml.com.cn/  
webservices/weatherwebservice.asmx";   
private static final String METHOD\_NAME ="getWeatherbyCityName";   
private static String SOAP\_ACTION ="http://WebXml.com.cn/  
getWeatherbyCityName";   
  
private String weatherToday;   
  
private Button okButton;   
private SoapObject detail;   
  
@Override   
publicvoid onCreate(Bundle savedInstanceState) {   
super.onCreate(savedInstanceState);   
setContentView(R.layout.main);   
okButton = (Button) findViewById(R.id.ok);   
  
okButton.setOnClickListener(new Button.OnClickListener() {   
publicvoid onClick(View v) {   
showWeather();   
}   
});   
}   
  
privatevoid showWeather() {   
String city ="武汉";   
getWeather(city);   
}   
  
@SuppressWarnings("deprecation")   
publicvoid getWeather(String cityName) {   
try {   
System.out.println("rpc------");   
SoapObject rpc =new SoapObject(NAMESPACE, METHOD\_NAME);   
System.out.println("rpc"+ rpc);   
System.out.println("cityName is "+ cityName);   
rpc.addProperty("theCityName", cityName);   
  
AndroidHttpTransport ht =new AndroidHttpTransport(URL);   
ht.debug =true;   
  
SoapSerializationEnvelope envelope =new SoapSerializationEnvelope(   
SoapEnvelope.VER11);   
  
envelope.bodyOut = rpc;   
envelope.dotNet =true;   
envelope.setOutputSoapObject(rpc);   
  
ht.call(SOAP\_ACTION, envelope);   
  
SoapObject result = (SoapObject) envelope.bodyIn;   
detail = (SoapObject) result   
.getProperty("getWeatherbyCityNameResult");   
  
System.out.println("result"+ result);   
System.out.println("detail"+ detail);   
Toast.makeText(WebService.this, detail.toString(),   
Toast.LENGTH\_LONG).show();   
parseWeather(detail);   
  
return;   
} catch (Exception e) {   
e.printStackTrace();   
}   
}   
  
privatevoid parseWeather(SoapObject detail)   
throws UnsupportedEncodingException {   
String date = detail.getProperty(6).toString();   
weatherToday ="今天："+ date.split("")[0];   
weatherToday = weatherToday +"\n天气："+ date.split("")[1];   
weatherToday = weatherToday +"\n气温："   
+ detail.getProperty(5).toString();   
weatherToday = weatherToday +"\n风力："   
+ detail.getProperty(7).toString() +"\n";   
System.out.println("weatherToday is "+ weatherToday);   
Toast.makeText(WebService.this, weatherToday,   
Toast.LENGTH\_LONG).show();   
  
}   
}