

7.4 Android 调用 Webservice

分类 [Android 基础入门教程](#)

Android 基础入门教程(Q群号：
153836263)

本节引言：

经过前面的学习，数据请求，数据解析，文件上传下载等，应该满足大家与服务器交互的基本需求了，而本节给大家介绍的Android调用WebService，其实这玩意有点类似于一些给我们提供原始数据API服务的数据平台，比如聚合数据！而WebService则用到了XML和SOAP，通过HTTP协议即可完成与远程机器的交互！嗯，不多说，开始本节内容~

1.WebService简介

Android平台调用WebService

①WebService的引入

我们都知道计手机硬件资源是有限的,对于一些复杂的数据处理,计算,通常都是部署在远程服务器上的,然后安卓手机作为对应的客户端;为了让android app与远程服务器进行交互,肯定是需要借助一些技术的了,各种高大上的技术,什么RMI,CORBA,一个都不会,没关系,我们现在会用Web Service平台就够了!那么,什么是WebService?

简单的说就是某些站点开放出来的服务,当然你也可以自己开发一个service,也就是一些方法,通过URL,指定某一个方法名,发出请求,站点的这个服务(方法),接收请求后,根据传入的参数做一些处理,然后将处理后的结果以XML形式返回给你,你的程序就解析这些XML参数,然后显示出来或做其他操作。
例如:很多大的站点提供有天气预报的webservice、查询某网站的数据的webservice,只要你发送请求过来,它就返回天气预报、某网站的数据,然后你把结果显示出来。

②主要采用的四个技术

XML	可扩展的标记语言。Xml是所有开发人员都熟悉的东西,不用多说,同时采用XML Schema。正因为WebService采用了Xml,才使得它可以跨越各种编程语言。
SOAP	简单对象访问协议。提供了标准的RPC方法来调用WebService。SOAP规范中定义了SOAP消息格式,以及怎样通过HTTP协议来使用SOAP。SOAP是基于XML语言和XSD标准的,其中XML是SOAP的数据编码方式。
WSDL	WSDL是一种基于XML的用于描述WebService及其操作、参数和返回值值得语言。因为是基于XML,一些IDE可以根据WebService来生成WSDL文档,又能通过导入WSDL文档,生成调用相应的WebService代码。
UDDI	UDDI用于在网上自动查找WebService,一旦WebService注册到UDDI,客户就可以很方便的查找和定位所需要的WebService。

③WebService的模型

服务提供者:把自己的服务注册到“服务注册中心”
服务请求者:到“服务注册中心”查找找到相应的服务,然后定位到“服务提供者”
服务提供者:开始为“服务请求者”提供相应的服务

PS:如果看完上面简介还不是很清楚的话,那么就算了,之前公司就用C#搭

1.0 Android基础入门教程

1.0.1 2015年最新Android基...

1.1 背景相关与系统架构分析

1.2 开发环境搭建

1.2.1 使用Eclipse + ADT + S...

1.2.2 使用Android Studio开...

1.3 SDK更新不了问题解决

1.4 Genymotion模拟器安装

1.5.1 Git使用教程之本地仓...

1.5.2 Git之使用GitHub搭建...

1.6 .9(九妹)图片怎么玩

1.7 界面原型设计

1.8 工程相关解析(各种文件...

1.9 Android程序签名打包

1.11 反编译APK获取代码&...

2.1 View与ViewGroup的概念

2.2.1 LinearLayout(线性布局)

2.2.2 RelativeLayout(相对布...

2.2.3 TableLayout(表格布局)

2.2.4 FrameLayout(帧布局)

2.2.5 GridLayout(网格布局)

2.2.6 AbsoluteLayout(绝对...

2.3.1 TextView(文本框)详解

2.3.2 EditText(输入框)详解

2.3.3 Button(按钮)与ImageB...

2.3.4 ImageView(图像视图)

2.3.5.RadioButton(单选按钮...

2.3.6 开关按钮ToggleButton...

2.3.7 ProgressBar(进度条)

2.3.8 SeekBar(拖动条)

2.3.9 RatingBar(星级评分条)

2.4.1 ScrollView(滚动条)

的一个WebService！本节我们并不讨论如何去搭建一个WebService，我们仅仅知道如何去获取WebService提供的服务，然后解析返回的XML数据，然后把相关数据显示到我们的Android设备上就好！

2.去哪里获取WebService服务

网上有很多提供WebService的站点，首先找到这些站点，然后获取相应的服务即可！这里选取WebXml和云聚36wu作为例子给大家讲解下，他们的官网：

webXml：http://www.webxml.com.cn/zh_cn/index.aspx

以前是免费的，不过都商业化了，很多服务都要收费，但是可以试用~改站点上提供了16个不同的Web服务，可以根据自己的需求，查询相应服务，调用不同的接口！

webXml的相关页面：



相关使用次数说明：

- 2.4.2 Date & Time组件(上)
- 2.4.3 Date & Time组件(下)
- 2.4.4 Adapter基础讲解
- 2.4.5 ListView简单实用
- 2.4.6 BaseAdapter优化
- 2.4.7 ListView的焦点问题
- 2.4.8 ListView之checkbox错...
- 2.4.9 ListView的数据更新问题
- 2.5.0 构建一个可复用的自定...
- 2.5.1 ListView Item多布局的...
- 2.5.2 GridView(网格视图)的...
- 2.5.3 Spinner(列表选项框)...
- 2.5.4 AutoCompleteTextVie...
- 2.5.5 ExpandableListView(...
- 2.5.6 ViewPager(翻转视图)...
- 2.5.7 Toast(吐司)的基本使用
- 2.5.8 Notification(状态栏通...
- 2.5.9 AlertDialog(对话框)详解
- 2.6.0 其他几种常用对话框基...
- 2.6.1 PopupWindow(悬浮框...
- 2.6.2 菜单(Menu)
- 2.6.3 ViewPager的简单使用
- 2.6.4 DrawerLayout(官方侧...
- 3.1.1 基于监听的事件处理机制
- 3.2 基于回调的事件处理机制
- 3.3 Handler消息传递机制浅析
- 3.4 TouchListener PK OnTo...
- 3.5 监听EditText的内容变化
- 3.6 响应系统设置的事件(Co...
- 3.7 AsyncTask异步任务
- 3.8 Gestures(手势)
- 4.1.1 Activity初学乍练
- 4.1.2 Activity初窥门径
- 4.1.3 Activity登堂入室
- 4.2.1 Service初涉
- 4.2.2 Service进阶
- 4.2.3 Service精通
- 4.3.1 BroadcastReceiver牛...
- 4.3.2 BroadcastReceiver庖...
- 4.4.1 ContentProvider初探

Email 电子邮件地址验证WEB服务

接口名称	免费用户	商业用户	说明
ValidateEmailAddress			24小时内不超过50次
ValidateEmailAddressPro			

IP地址来源搜索WEB服务

接口名称	免费用户	商业用户	说明
getCountryCityByIp			免费用户24小时内不超过200次
getGeolPContext			

中国股票行情数据WEB服务

接口名称	免费用户	商业用户	说明
getStockImageByCode			免费用户24小时内不超过250次；数据延时0-30秒，为商业用户添加图片LOGO水印
getStockImageByteByCode			同上
getStockImage_kByCode			同上
getStockImage_kByteByCode			同上
getStockInfoByCode			免费用户24小时内不超过250次；数据延时0-30秒

中国股票行情分时走势预览缩略图WEB服务

接口名称	免费用户	商业用户	说明
getSmallImage			免费用户24小时内不超过250次
getSmallImageByte			同上

中国邮政编码 <-> 地址信息双向查询/搜索WEB服务

接口名称	免费用户	商业用户	说明
getAddressByZipCode			免费用户24小时内不超过60次

云聚36wu : <http://www.36wu.com/Service>

同样也提供了很多的服务,很多手机的app都是用的这里的接口,比如彩虹公交,手机天气等 不过,这个也是要收费的==,可以试用,不过只能一小时内发送20次请求; 点击申请使用,获得key就可以了!两者随便选一个吧!

首页数据与接口经典案例支持服务文档说明关于我们申请试用

数据与接口

累计调用次数：24,211,12

全国天气查询

快递/物流跟踪

火车票/时刻查询

翻译

IP地址查询

手机归属地

全国公交查询

全国车辆违章查询

身份证查询

ISBN信息查询

全国POI信息

影片/资讯查询

http://blog.csdn.net/coder_pig

3.第三方jar包的准备

首先如果想在Android平台上调用WebService需要依赖于第三方类库:ksoap2 而在Android平台上,使用的是ksoap2 Android,一个高效,轻量级的SOAP开发包!

jar包下载地址：<https://code.google.com/p/ksoap2-android/wiki/HowToUse?tm=2>


天朝可能上不去，这里提供两个百度网盘的链接供大家下载使用：

- 4.4.2 ContentProvider再探...
- 4.5.1 Intent的基本使用
- 4.5.2 Intent之复杂数据的传递
- 5.1 Fragment基本概述
- 5.2.1 Fragment实例精讲—...
- 5.2.2 Fragment实例精讲—...
- 5.2.3 Fragment实例精讲—...
- 5.2.4 Fragment实例精讲—...
- 5.2.5 Fragment实例精讲—...
- 6.1 数据存储与访问之——文...
- 6.2 数据存储与访问之——S...
- 6.3.1 数据存储与访问之——...
- 6.3.2 数据存储与访问之——...
- 7.1.1 Android网络编程要学...
- 7.1.2 Android Http请求头与...
- 7.1.3 Android HTTP请求方...
- 7.1.4 Android HTTP请求方...
- 7.2.1 Android XML数据解析
- 7.2.2 Android JSON数据解析
- 7.3.1 Android 文件上传
- 7.3.2 Android 文件下载（1）
- 7.3.3 Android 文件下载（2）
- 7.4 Android 调用 Webservice
- 7.5.1 WebView(网页视图)基...
- 7.5.2 WebView和JavaScrip...
- 7.5.3 Android 4.4后WebVie...
- 7.5.4 WebView文件下载
- 7.5.5 WebView缓存问题
- 7.5.6 WebView处理网页返...
- 7.6.1 Socket学习网络基础准备
- 7.6.2 基于TCP协议的Socket...
- 7.6.3 基于TCP协议的Socket...
- 7.6.4 基于UDP协议的Socke...
- 8.1.1 Android中的13种Draw...
- 8.1.2 Android中的13种Draw...
- 8.1.3 Android中的13种Draw...
- 8.2.1 Bitmap(位图)全解析 P...
- 8.2.2 Bitmap引起的OOM问题
- 8.3.1 三个绘图工具类详解
- 8.3.2 绘图类实战示例

2.54版本：[ksoap2-android 2.54.jar](#)

3.30版本：[ksoap2-android 3.30.jar](#)

如果所幸你能进入jar包的下载地址的话,那么你会看到下面的界面:



ksoap2-android

A lightweight and efficient SOAP library for the Android platform.

[Project Home](#)[Downloads](#)[Wiki](#)[Issues](#)[Source](#)

General

- [How to use](#)
- [Project News](#)
- [Projects Using Ksoap2Android](#)
- [Coding Tips and Tricks](#)
- [Cookies](#)
- [You need help?](#)
- [Useful links](#)
- [History](#)
- [Why contribute?](#)
- [Mailing List](#)
- [License Information](#)

Contributors

- [Obtaining the source](#)
- [How to contribute](#)
- [Working with the source code](#)

HowToUse

Find out how to use this library in your Android application

How to Use

Apache Maven

To make use of this library in a Maven project, add the following to your project's POM:

```
<project>
  <dependencies>
    <dependency>
      <groupId>com.google.code.ksoap2-android</groupId>
      <artifactId>ksoap2-android</artifactId>
      <version>3.3.0</version>
    </dependency>
  </dependencies>
</project>
```

and configure a proxy repository in your repository manager using the url

```
http://ksoap2-android.googlecode.com/svn/m2-repo
```

If you are not using a repository manager like [Sonatype Nexus](#), you should start doing so now.

As a stop gap solution you could add the repository to a profile in your settings.xml or pom.xml:

```
<repositories>
  <repository>
    <id>google-code-ksoap2</id>
    <url>http://ksoap2-android.googlecode.com/svn/m2-repo</url>
  </repository>
</repositories>
```

Contact [simpligility technologies](#) for consulting and training for [Sonatype Nexus](#).

Further libraries as part of the project are also published as part of the build and can be declared as dependencies as well, if needed.

Standard Android Ant-based build and Eclipse/ADT

To make use of this library in a non-Maven project, follow the instructions in the [Android Developer's Guide](#) on how to [Add an External Library](#) to your project.

You will need to add a ksoap2-android and all required transitive dependencies to the build path. Luckily the Maven build of the project produces a nice bundle of all these jars in one big file.

The different version of these files are available at <http://code.google.com/p/ksoap2-android/source/browse/#svn/m2-repo/com/google/code/ksoap2-android/ksoap2-android-assembly>

To download a file from there, right click on "View raw file" and select "Save Link as" (this label differs for different browsers) and you will get the full jar downloaded.

The latest release artifact would be available at

<http://code.google.com/p/ksoap2-android/source/browse/m2-repo/com/google/code/ksoap2-android/ksoap2-android-assembly/3.3.0/ksoap2-android-assembly-3.3.0-jar-with-dependencies.jar>

with a direct download url of

<http://ksoap2-android.googlecode.com/svn/m2-repo/com/google/code/ksoap2-android/ksoap2-android-assembly/3.3.0/ksoap2-android-assembly-3.3.0-jar-with-dependencies.jar>

Some users seem to experience download problems with IE. Just try a decent browser or download with a command line tool like wget

```
wget http://ksoap2-android.googlecode.com/svn/m2-repo/com/google/code/ksoap2-android/ksoap2-android-assembly/3.3.0/ksoap2-android-assembly-3.3.0-jar-with-dependencies.jar
```

After the manual download confirm the size of the downloaded file since many users seem to be experiencing problems.

8.3.3 Paint API之—— Mask...

8.3.4 Paint API之—— Xferm...

8.3.5 Paint API之—— Xferm...

8.3.6 Paint API之—— Xferm...

8.3.7 Paint API之—— Xferm...

8.3.8 Paint API之—— Xferm...

8.3.9 Paint API之—— Color...

8.3.10 Paint API之—— Colo...

8.3.11 Paint API之—— Colo...

8.3.12 Paint API之—— Path...

8.3.13 Paint API之—— Sha...

8.3.14 Paint几个枚举/常量值...

8.3.15 Paint API之——Type...

8.3.16 Canvas API详解(Part 1)

8.3.17 Canvas API详解(Part...

8.3.18 Canvas API详解(Part...

8.4.1 Android动画合集之帧...

8.4.2 Android动画合集之补...

8.4.3 Android动画合集之属...

8.4.4 Android动画合集之属...

9.1 使用SoundPool播放音...

9.2 MediaPlayer播放音频与...

9.3 使用Camera拍照

9.4 使用MediaRecord录音

10.1 TelephonyManager(电...

10.2 SmsManager(短信管理...

10.3 AudioManager(音频管...

10.4 Vibrator(振动器)

10.5 AlarmManager(闹钟服务)

10.6 PowerManager(电源服...

10.7 WindowManager(窗口...

10.8 LayoutInflater(布局服务)

10.9 WallpaperManager(壁...

10.10 传感器专题(1)——相...

10.11 传感器专题(2)——方...

10.12 传感器专题(3)——加...

10.12 传感器专题(4)——其...

10.14 Android GPS初涉

11.0 《2015最新Android基...

4.获取相关的一些参数

首先找到我们需要获取的服务，然后记录相关的参数：

Namespace(命名空间),SoapAction以及**URL**就不用说了,其他参数这样找:

国内手机号码归属地查询WEB服务 通讯和通信Endpoint: <http://webservice.webxml.com.cn/WebServices/MobileCodeWS.asmx>Disco: <http://webservice.webxml.com.cn/WebServices/MobileCodeWS.asmx?disco>WSDL: <http://webservice.webxml.com.cn/WebServices/MobileCodeWS.asmx?wsdl>

国内手机号码归属地查询WEB服务，提供最新的国内手机号码归属地数据，每月更新。包括最新的电信天翼189号段和最新移动152号段、TD-SCDMA188号段。数据更全更准确，是目前国内最新最全的手机号码数据库！

**2500多个城市天气预报 WEB服务** 公用事业Endpoint: <http://webservice.webxml.com.cn/WebServices/WeatherWS.asmx>Disco: <http://webservice.webxml.com.cn/WebServices/WeatherWS.asmx?disco>WSDL: <http://webservice.webxml.com.cn/WebServices/WeatherWS.asmx?wsdl>

2500多个城市天气预报Web服务，包含2400个以上中国城市和100个以上国外城市天气预报数据。数据每2.5小时左右自动更新一次，准确可靠。为让更多的开发人员学习WEB服务开发，此服务支持免费用户使用。为支持多种平台开发，此WEB服务接口提供了多种返回类型可选择。

http://blog.csdn.net/coder_pig

比如我们这里找的是天气的查询参数，点进去我们可以看到这样一个参数文档：

WeatherWS

[WebXml.com.cn](http://www.webxml.com.cn) 2400多个城市天气预报Web服务，包含2300个以上中国城市和100个以上国外城市。使用本站 WEB 服务请注明或链接本站：<http://www.webxml.com.cn/> 感谢大家的支持！

[接口帮助文档](#) [部分城市介绍和气候背景](#) [部分城市图片](#) [天气现象和图例](#)

支持下列操作。有关正式定义，请查看[服务说明](#)。

- [getRegionCountry](#)

获得国外国家名称和与之对应的ID

输入参数：无，返回数据：一维字符串数组。

- [getRegionDataset](#)

获得中国省份、直辖市、地区；国家名称（国外）和与之对应的ID

输入参数：无，返回数据：DataSet。

- [getRegionProvince](#)

获得中国省份、直辖市、地区和与之对应的ID

输入参数：无，返回数据：一维字符串数组。

- [getSupportCityDataset](#)

获得支持的城市/地区名称和与之对应的ID

输入参数：theRegionCode = 省市、国家ID或名称，返回数据：DataSet。

- [getSupportCityString](#)

获得支持的城市/地区名称和与之对应的ID

输入参数：theRegionCode = 省市、国家ID或名称，返回数据：一维字符串数组。

- [getWeather](#)

获得天气预报数据

输入参数：城市/地区ID或名称，返回数据：一维字符串数组。

比如这里我们需要的是天气查询部分的功能：

WeatherWS

单击[此处](#)，获取完整的操作列表。

getWeather

获得天气预报数据

输入参数: **城市/地区ID或名称** 返回数据: 一维字符串数组。

测试

若要使用 HTTP POST 协议对操作进行测试，请单击“调用”按钮。

参数	值
theCityCode:	<input type="text"/>
theUserID:	<input type="text"/>

SOAP 1.1

以下是 SOAP 1.2 请求和响应示例。所显示的占位符需替换为实际值。

```
POST /WebServices/WeatherWS.asmx HTTP/1.1
Host: webservice.webxml.com.cn
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://WebXml.com.cn/getWeather" SoapAction

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://
<soap:Body>
  <getWeather xmlns="http://WebXml.com.cn/"> Namespace(命名空间)
    <theCityCode>string</theCityCode>
    <theUserID>string</theUserID>
  </getWeather>
</soap:Body>
</soap:Envelope>
```

先把框住的SoapAction和NameSpace拷贝下来！当然我们可以在这个页面测试，另外 我们是免费用户，id可以不填直接跳过，输入后点击调用按钮会打开这样一个页面：

This XML file does not appear to have any style informatio

```
▼<ArrayOfString xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" x
  <string>广东 珠海</string>
  <string>珠海</string>
  <string>2417</string>
  <string>2015/09/10 10:32:11</string>
  <string>今日天气实况: 气温: 29℃; 风向/风力: 东风 1级; 湿度: 72%</st
  <string>空气质量: 暂无; 紫外线强度: 很强</string>
▼<string>
  太阳镜指数: 很必要。建议佩戴透射比2级且UV400的遮阳镜。 穿衣指数: 忌
  霜。 感冒指数: 少发。感冒机率较低, 避免长期处于空调屋中。 空气污染
</string>
<string>9月10日 多云</string>
<string>27℃/31℃</string>
<string>无持续风向微风</string>
<string>1.gif</string>
<string>1.gif</string>
<string>9月11日 多云</string>
<string>26℃/31℃</string>
<string>无持续风向微风</string>
<string>1.gif</string>
<string>1.gif</string>
<string>9月12日 多云</string>
<string>25℃/31℃</string>
<string>无持续风向微风转北风3-4级</string>
<string>1.gif</string>
<string>1.gif</string>
<string>9月13日 小雨转多云</string>
<string>26℃/31℃</string>
<string>北风3-4级转无持续风向微风</string>
<string>7.gif</string>
<string>1.gif</string>
<string>9月14日 多云</string>
<string>27℃/32℃</string>
<string>无持续风向微风</string>
<string>1.gif</string>
<string>1.gif</string>
</ArrayOfString>
```


嘿嘿，这里就是返回的XML，而我们要做的也就是解析这样一个XML，另外这里的.gif代表的是天气图标！

同理，我们再把归属地查询的看下SoapAction，NameSpace以及相关参数mark下！

MobileCodeWS

单击[此处](#)，获取完整的操作列表。

getMobileCodeInfo

获得国内手机号码归属地省份、地区和手机卡类型信息

输入参数：mobileCode = 字符串（手机号码，最少前7位数字），userID = 字符串（商业用户ID） 免费用户为空字符串；

测试

若要使用 HTTP POST 协议对操作进行测试，请单击“调用”按钮。

参数	值
mobileCode:	13798983314
userID:	

调用

SOAP 1.1

以下是 SOAP 1.2 请求和响应示例。所显示的占位符需替换为实际值。

```
POST /WebServices/MobileCodeWS.asmx HTTP/1.1
Host: webservice.webxml.com.cn
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://WebXml.com.cn/getMobileCodeInfo"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getMobileCodeInfo xmlns="http://WebXml.com.cn/">
      <mobileCode>string</mobileCode>
      <userID>string</userID>
    </getMobileCodeInfo>
  </soap:Body>
</soap:Envelope>
```

以及返回后的XML数据：

This XML file does not appear to have any style information associated with it. The file may have been moved.

```
<string xmlns="http://WebXml.com.cn/">13798983314 : 广东 珠海 广东移动动感地带卡</string>
```

5.注册并启用相关WEB服务

反馈

会员专区

会员专区首页

欢迎登录会员专区

②登陆后来到会员界面

庄培杰

会员名: [redacted]

WEB服务 用户ID: fbcd3137f2104e01a8732aaccd19f4dd [如何使用?](#)

WEB服务 测试数: 4次 *

预存款余额: ¥ 0.00 元 (RMB)

称呼: 先生

注册日期: 2014年09月22日 11:52 星期一

上次登录: 2014年09月22日 20:50 星期一

登录次数: 4次

电子邮件: 779878443@qq.com

网站:

帮助

允许你使用它的服务的次数

这个就是id,或者说密钥
如果没密钥,你也可以使用服务,但是没数据==

* 每次测试数可用于WEB服务测试 5 天, 每在线预存人民币 ¥ 100.00 元增加 1 次测试数。

http://blog.csdn.net/code_pig

点击我的Web服务器，然后点击试用，WebXML给我们提供了五天的免费试用，我们把需要的两个服务器开启！

我的WEB服务

WEB服务	服务截止	试用	购买	基本价格*
中国邮政编码 <=> 地址双向查询	N/A	试用	购买	¥ 120.00 *
IP地址来源搜索	N/A	试用	购买	¥ 96.00 *
2400多个城市天气预报	15/09/15 11:00	试用	购买	¥ 168.00 *
国内飞机航班时刻表	N/A	试用	购买	¥ 168.00 *
中国电视节目预告	N/A	试用	购买	¥ 168.00 *
火车时刻表	N/A	试用	购买	¥ 144.00 *
中国开放式基金数据	N/A	试用	购买	¥ 72.00 *
国内移动电话号码段归属地	15/09/15 11:00	试用	购买	¥ 80.00 *
腾讯QQ在线状态	N/A	试用	购买	¥ 96.00 *
Email 电子邮件地址验证	N/A	试用	购买	¥ 120.00 *

您的WEB 服务 测试数: 3 次

* 基本价格为购买此服务的一个IP地址、每天查询10000次、一个月价格（单位：人民币元）。具体价格根据您购买的服务日期长短会不同，详情请点击购买按钮 [购买] 后查看。网上支付购买WEB服务费用不含税点、快递等其他服务费用，如有更多需求（如：大数量、多IP访问，索要发票、合同等）请先联系我们，点击[这里](#)下载服务合同

好的，记得mark下我们自己的key哦~

6.调用WebService的代码示例

嗯，接下来我们来写代码验证调用WebService的流程：

运行效果图：



PS:这个号码是以前的号码==，别尝试拨打，已经换人了~ 另外天气服务好像有写问题，有时并不能获取到，估计是WebXml做的一些限制，毕竟试用...

实现代码：

```

public class MainActivity extends AppCompatActivity implements
View.OnClickListener {

    private EditText edit_param;
    private Button btn_attribution;
    private Button btn_weather;
    private TextView txt_result;

    private String city;
    private String number;
    private String result;

    //定义获取手机信息的SoapAction与命名空间,作为常量
    private static final String AddressnameSpace = "http://WebX
ml.com.cn/";
    //天气查询相关参数
    private static final String Weatherurl = "http://webservice
.webxml.com.cn/WebServices/WeatherWS.asmx";
    private static final String Weathermethod = "getWeather";
    private static final String WeathersoapAction = "http://Web
Xml.com.cn/getWeather";
    //归属地查询相关参数
    private static final String Addressurl = "http://webservice
.webxml.com.cn/WebServices/MobileCodeWS.asmx";
    private static final String Addressmethod = "getMobileCodeI
nfo";
    private static final String AddresssoapAction = "http://Web
Xml.com.cn/getMobileCodeInfo";

    //定义一个Handler用来更新页面:
    private Handler handler = new Handler() {
        public void handleMessage(Message msg) {
            switch (msg.what) {
                case 0x001:
                    txt_result.setText("结果显示: \n" + result);
                    Toast.makeText(MainActivity.this, "获取天气
信息成功", Toast.LENGTH_SHORT).show();
                    break;
                case 0x002:
                    txt_result.setText("结果显示: \n" + result);
                    Toast.makeText(MainActivity.this, "号码归属
地查询成功", Toast.LENGTH_SHORT).show();
                    break;
            }
        }
    };

    @Override

```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    bindViews();
}

private void bindViews() {
    edit_param = (EditText) findViewById(R.id.edit_param);
    btn_attribution = (Button) findViewById(R.id.btn_attribution);
    btn_weather = (Button) findViewById(R.id.btn_weather);
    txt_result = (TextView) findViewById(R.id.txt_result);
    btn_attribution.setOnClickListener(this);
    btn_weather.setOnClickListener(this);
}

@Override
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.btn_weather:
            new Thread() {
                @Override
                public void run() {
                    getWether();
                }
            }.start();
            break;
        case R.id.btn_attribution:
            new Thread(new Runnable() {
                public void run() {
                    getland();
                }
            }).start();
            break;
    }
}

//定义一个获取某城市天气信息的方法:
public void getWether() {
    result = "";
    SoapObject soapObject = new SoapObject(AddressnameSpace, Weathermethod);
    soapObject.addProperty("theCityCode:", edit_param.getText().toString());
    soapObject.addProperty("theUserID", "dbdf1580476240458784992289892b87");
    SoapSerializationEnvelope envelope = new SoapSerializationEnvelope(SoapEnvelope.VER11);
    envelope.bodyOut = soapObject;
    envelope.dotNet = true;
    envelope.setOutputSoapObject(soapObject);
    HttpTransportSE httpTransportSE = new HttpTransportSE(Weatherurl);

```



```

        System.out.println("天气服务设置完毕,准备开启服务");
        try {
            httpTransportSE.call(WeathersoapAction, envelope);
//            System.out.println("调用WebService服务成功");
        } catch (Exception e) {
            e.printStackTrace();
//            System.out.println("调用WebService服务失败");
        }

        //获得服务返回的数据,并且开始解析
        SoapObject object = (SoapObject) envelope.bodyIn;
        System.out.println("获得服务数据");
        result = object.getProperty(1).toString();
        handler.sendEmptyMessage(0x001);
        System.out.println("发送完毕,textview显示天气信息");
    }

    //定义一个获取号码归属地的方法:
    public void getland() {
        result = "";
        SoapObject soapObject = new SoapObject(AddressnameSpace
, Addressmethod);
        soapObject.addProperty("mobileCode", edit_param.getText
().toString());
        soapObject.addProperty("userid", "dbdf15804762404587849
92289892b87");
        SoapSerializationEnvelope envelope = new SoapSerializat
ionEnvelope(SoapEnvelope.VER11);
        envelope.bodyOut = soapObject;
        envelope.dotNet = true;
        envelope.setOutputSoapObject(soapObject);
        HttpTransportSE httpTransportSE = new HttpTransportSE(A
ddressurl);
//        System.out.println("号码信息设置完毕,准备开启服务")
;
        try {
            httpTransportSE.call(AddresssoapAction, envelope);
//            System.out.println("调用WebService服务成功");
        } catch (Exception e) {
            e.printStackTrace();
//            System.out.println("调用WebService服务失败");
        }

        //获得服务返回的数据,并且开始解析
        SoapObject object = (SoapObject) envelope.bodyIn;//Syst
em.out.println("获得服务数据");
        result = object.getProperty(0).toString();//System.out.
println("获取信息完毕,向主线程发信息");
        handler.sendEmptyMessage(0x001);
        //System.out.println("发送完毕,textview显示天气信息");
    }

```

```
}
}
```

另外，别忘了导包和Internet的权限！

```
<uses-permission android:name="android.permission.INTERNET"/>
```

参考代码下载：

WebServiceDemo.zip：[下载 WebServiceDemo.zip](#)

本节小结：

好的，本节关于Android端如何去使用这个WebService就讲解到这里，下一节我们来学习一个 类似于浏览器的Android控件——WebView，敬请期待~谢谢~！



坚持1个月 看美剧不用字幕

每天45分钟 30天见证你的英语奇迹

立即行动 >



在线实例	字符集&工具	最新更新	站点信息
<div><ul style="list-style-type: none">· HTML 实例· CSS 实例· JavaScript 实例· Ajax 实例· jQuery 实例· XML 实例· Java 实例</div>	<div><ul style="list-style-type: none">· HTML 字符集设置· HTML ASCII 字符集· HTML ISO-8859-1· HTML 实体符号· HTML 拾色器· JSON 格式化工具</div>	<div><ul style="list-style-type: none">· 设置 SSH 通过密...· CSS all 属性· Px、Em 换算工具· px,pt,em换算表· px、em、rem 区别...· Viewport 模板· 移动WEB前端开发...</div>	<div><ul style="list-style-type: none">· 意见反馈· 免责声明· 关于我们· 文章归档</div>
<div><div>关注微信</div><div></div><div>Copyright © 2013-2015 菜鸟教程 runoob.com All Rights Reserved. 备案号：闽ICP备15012807号-1</div></div>			