

Android 手机访问服务器的一种数据交互方法

刘 平

(西安财经学院 网络与现代教育技术中心, 陕西 西安 710061)

摘要:通过对 Android 手机访问 Web 服务器的 API 的分析和实践,提出了一种解决手机终端和 Web 服务器交互提取各种类型数据的清晰有效的方法,扩展了 Android 平台在访问 Web 服务器进行交互时的解析数据能力,给移动商务软件开发人员在开发数据交互实践过程提供了参考。

关键词:Android; 手机终端; 3G; 数据流; 过滤流; Servlet; SSH 架构; POST 方法; GET 方法; XML; 数据库

中图分类号: TP393.02

文献标识码: A

文章编号: 1674-6236(2010)09-0096-03

Effective data interactive method from Android cell phone accessing Web app server

LIU Ping

(Information & Modern Education Technique Center, Xi'an University of Finance and Economics, Xi'an 710061, China)

Abstract: Through the analysis and practice of Android API access to Web App Server, this paper puts forward a data interactive method, which is clearly and effectively used to get the data of the different types from Web App Server, and extend the data interactive ability between the cell phone and Web App Server, and also provide reference for those who is engaged in mobile business software development.

Key words: Android; cell phone; 3G; data stream; filter flow; servlet; SSH; POST; GET; XML; database

为了让手机用户能够在任何时候 (Anytime)、任何地方 (Anywhere)、以任何方式 (Any way) 查询企业所提供的服务,一种高效的办法就是将应用系统的功能拓展到手机终端,手机通过移动网和互联网访问企业 Web 网站并处理各项业务 (移动商务系统)。从市场的角度来讲,将企业提供的服务通过现代的先进技术拓展到手机终端,是一项具有极大潜在的市场需求。由此可见,能够给手机终端提供有效、便捷和清晰地访问 Web 服务器数据的方法在实际中非常有用。随着 3G 网络的使用,移动终端不再仅是通讯网络的终端,还将成为互联网的终端。因此,移动终端的应用软件和需要的服务将会有很大的发展空间。Google 为此于 2007 年 11 月推出了一个专为移动设备设计的软件平台——Android,它是一套真正意义上的开放性移动设备综合平台,其包括操作系统、中间件和一些关键的平台应用 (见图 1)。Android 最大特点在于它是一个开放的体系架构,具有非常好的开发和调试环境,而且还支持各种可扩展的用户体验,Android 里面包含非常丰富的图形组建,多媒体支持功能和非常强大的浏览器^[1]。Android 平台的开放性等特点既能促进技术 (包括平台自身) 的创新,又有助于降低开发成本,还可以使运营商能非常方便地制定特色化产品,因此,它具有很大的市场发展潜力。谷歌的 Android 手机操作系统在发布以后,每年的 Android 手机上市速度惊人,预计在不久的将来大有占据手机大份额市场的可能,An-

droid 手机应用程序是以 XML 文件作为其界面和数据的描述文件,对于每一个组件和数据都有唯一的 R.java 文件标识,根据这些标识来加载和确定组件对象,进而通过事件来驱动事务,Android 的重要技术是提供了一套 API 来解析这些 XML 描述文件,编写的程序紧凑、数据安全,界面丰富,交互模式多样化,应用程序架构清晰,同时也比较抽象,对开发人员要求比较高。但是由于 Android 发布的是新技术,难免在实际应用中还存在发展的空间,有些方面还需要进一步完善,需要在市场中不断开发新的技术结合 Android 提供的 API,将开发的应用程序更加安全、完美和有效。为此,笔者经过对 Android 提供的 API 访问 Web 服务器进行数据通信进行了系统研究,认为 Android 提供的技术在框架上很大的优势,但在具体取数据和发数据的细节上还需要增加新的技术来弥补。为此,本文笔者就开发 Android 手机软件过程中创建的数据交换和提取不同类型数据的方法和过程进行了阐述,为开发者提供参考。

1 Android 手机访问 Web 服务器技术架构

大多数中间件技术开发人员熟悉如何用计算机浏览器访问互联网,浏览器的主要作用是给互联网的 Web 服务器提交数据、验证数据和解析显示数据。其工作原理是通过 Http 协议,提交数据用 GET 或者 POST 方法,客户端的数据通过

收稿日期:2010-07-10

稿件编号:201007036

基金项目:国家科技部科技型中小企业技术创新基金资助项目 ((CA04)04C26216110880)

作者简介:刘平 (1959—),女,陕西佳县人,讲师。研究方向:嵌入式手机通信软件。

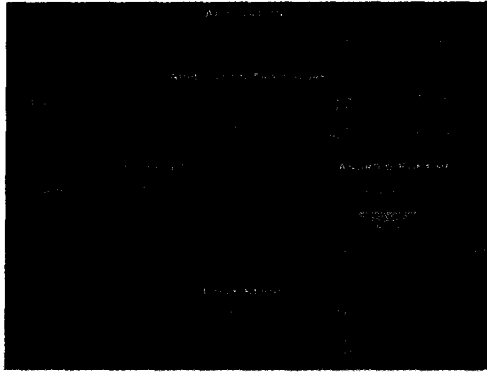


图1 Android系统架构图

Fig. 1 Android system architecture

浏览器网页提交给 Web 应用服务器,应用服务器通过 Web 页面接收各种不同类型数据,将数据通过服务器的 Servlet 子类 HttpServlet 对象提交给服务器端处理逻辑,服务器逻辑将接收到的数据按照客户端的要求在 Web 服务器端进行运算,再将运算的结果返回给客户端浏览器进行解析和显示。这里关键是 Web 服务器端的 Servlet,它是服务器和客户端交互数据的服务器端端口,即所有客户端的数据都要通过 Servlet 提交给 Web 服务器,所有要返回给客户端的数据都要通过 Web 服务器端的 Servlet 响应给客户端。在实际应用的服务器端,创建 Servlet 的子类对象 HttpServlet^[2](HttpServletRequest 和 HttpServletResponse),分别用来接收客户端的数据和将数据返回给客户端。

现在的问题就在 Android 手机终端,如何能够把客户的数据通过手机界面提交给 Web 服务器? Android 手机终端访问 Web 服务器的技术架构是怎样的?又如何能够在手机终端把 Web 服务器响应的数据按照服务器的不同数据类型恢复原型?这就是下面要解决的问题。

1.1 手机客户端向 Web 应用服务器发送请求信息

以实际开发的 Android 手机终端登录功能为案例介绍如何利用 Sun 公司提供的 java 数据流和过滤流结合 Android 提供的 API 提取 Web 服务器端不同类型的数据。

1)首先引用所需要的各种 API 在 Android 手机终端开发应用程序时,首先需要利用 Google 和 Sun 公司提供的必要类包(API),包括 http 通信协议下的类 org.apache.http.* 手机端类 org.apache.http.client.* 消息类 org.apache.http.message.* 数据流类 java.io.* 交互界面类 java.util.*, android.app.Activity 操作描述类 android.content.* 键值处理类 android.os.Bundle; 日志类 android.util.Log 显示组件类 android.view.* 和界面类 android.widget.* 等。

2)定义访问网站的 URL

```
public class UriAPI {
    public static final String TTPCustomer
        ="http://192.168.157.100:8028/JSHWDemo/servlet/
        CustomerServlet";
}
```

3)创建 Android 程序界面类 创建一个手机界面,需要

继承 Android 提供的 Activity 类,在该类中创建提交给 Web 服务器的数据的 HttpPost 对象,覆盖 Activity 类的方法 onCreate(Bundle b),创建事件监听器 OnClickListener 对象,在其中实现 onClick(View v)方法,再把该事件监听器注册到相应的事件源 loginButton 上。主要代码如下:

```
public class LoginActivity extends Activity {
    private static final int REQUEST_CODE = 2;
    HttpPost httpRequest=new HttpPost(UriAPI.HTTPCustomer);
    Button loginButton;
    Button cancel;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        loginButton=(Button) findViewById(R.id.login);
        //注册监听器对象
        loginButton.setOnClickListener(new OnClickListener(){
            CharSequence username="";
            CharSequence password="";
            public void onClick(View v)
            {
                //准备需要提交的数据,从手机界面抓取下来
                EditText ev_username=(EditText) findViewById(R.id.
                username);
```

4)创建 HttpPost 对象 HttpPost 对象的主要作用是将在 Android 手机端的数据提交给 Web 应用服务器,为此 Android 提供了很好的 API,这里就利用 Android 和 Sun 公司提供的这些类把需要提交给服务器的数据(NameValuePair 对象)保存到一个 List 对象中,主要代码如下:

```
if(! username.equals("")&&! password.equals("")){
    //创建 HttpPost 对象
    HttpPost httpRequest =new HttpPost (UriAPI.HTTPCustomer);
    List<NameValuePair> params=new ArrayList<NameValuePair>();
    params.add (new BasicNameValuePair ("username", username.toString()));
    params.add (new BasicNameValuePair ("password", password.toString()));
    params.add(new BasicNameValuePair("flag","0"));
```

5)设置数据编码方式 Android 手机在将准备好的数据发送到 Web 服务器之前,需要对数据的编码进行规定,这样服务器在接收到这些数据后,就会根据发送过去的文字编码处理和显示,代码如下:

```
try{
    httpRequest.setEntity (new UrlEncodedFormEntity (params,
    HTTP.UTF_8));
```

6)提交HttpPost对象和获取服务器HttpResponse响应数据 完成前面的工作,就该将数据提交给Web应用服务器了,在应用服务接收Android手机提交的数据后,经过处理,将要返回(响应)给Android手机的数据打包到一个HttpResponse对象中,发回给手机,代码如下:

```
    HttpResponse httpResponse=new DefaultHttpClient().execute(httpRequest);
}
}catch(ClientProtocolException e){e.printStackTrace();}
catch(UnsupportedEncodingException e) {e.printStackTrace();}
();}
catch (IOException e) {e.printStackTrace();}
```

到此,Android平台的通信任务已完成,即将Web服务器相应过来的数据交给了httpResponse对象,剩下的工作就是如何将Web服务器响应过来的数据从httpResponse对象中取出。

1.2 手机终端提取服务器端的响应数据

1)获取响应服务器的数据 从Web服务器响应到手机终端的数据一般打包在一个字节数组中,关键是这个字节数组中可能包含了各种不同的数据类型,为此,需要采取Java数据流和过滤流的方法从字节数据中取出各种类型的数据,这是本文介绍的关键技术,代码如下:

```
if(httpResponse.getStatusLine().getStatusCode()==200)
{
    //利用字节数组流和包装的绑定技术
    byte[] data = new byte[2048];
    //先把从服务端来的数据转化成字节数组
    data =EntityUtils.toByteArray ((HttpEntity) httpResponse.
    getEntity());
```

//再创建字节数组输入流对象

```
ByteArrayInputStream bais = new ByteArrayInputStream(data);
```

//绑定字节流和数据包装流

```
DataInputStream dis = new DataInputStream(bais);
```

将字节数组中的数据还原成原来的各种数据类型,代码如下:

```
user=new String(dis.readUTF());
```

到此手机终端从Web服务器获取了所有类型的数据,剩下的任务就是将这些数据进一步交给Android手机终端View组件^[9],在手机界面上显示给用户,这方面的内容不再赘述。

2 服务器端接收和响应Android手机终端请求

在Android手机终端发送请求后,接收请求的Web服务

器端会根据请求的数据信息进行运算,然后把运算结果响应给手机客户端。下面介绍Web服务器如何实现这一过程的。

2.1 采用http的Post方法接收数据

1)引用所需API 在Web端开发应用程序时,首先需要利用Sun公司提供的必要类包(API),包括数据流类java.io.* 交互界面类java.util.* 数据通信端口类javax.servlet.* 通信协议类javax.servlet.http.* 用Spring架构通过XML文件注入数据对象类org.springframework.web.context.*,org.springframework.web.context.support.*service.CustomerService等。

2)创建Servlet类数据端口 创建Servlet对象就是创建了Web应用服务器和连接互联网上所用客户端通信设备的数据接收和发送端口,主要代码如下:

```
public class CustomerServlet extends HttpServlet {
    public void doPost (HttpServletRequest request,HttpServletResponse response)
    throws ServletException, IOException {
        String username=request.getParameter("username");
        String password=request.getParameter("password");
        response.setContentType(CONTENT_TYPE);
```

3)访问持久层检索数据 这里应用已经完善的Spring管理事务的对象方法来获取持久层的数据,主要代码如下:

```
try{
    customer=cusDao.login(username,password);
    if(customer!=null){//登录成功
        output.writeUTF(customer.getUsername());
    }
    else{//登录失败
        output.writeUTF("No");
    }
}catch(Exception e){e.printStackTrace();}
```

4)创建输出流数据流对象 这里创建的输出流对象主要是把响应的数据发送到手机端,由于可能是不同类型的数据,所以采用DataOutputStream对象来处理,用输出过滤流对象^[10]将不同类型的数据返回手机终端,主要代码如下:

```
DataOutputStream output = new DataOutputStream (
    response.getOutputStream());
    output.writeUTF("服务器端数据:用户名:"+username);
    output.writeUTF("服务器端数据:密码:"+password);
    output.writeInt(1);
    output.close();
}
```

2.2 配置服务器端组件集成文件(XML)

Web应用服务器端用到的是SSH架构^[11]技术,服务器的业务逻辑已经成熟,只需要将Spring的业务通过前述的Servlet进行封装,就可以将所有的Spring管理的事物业务通过Servlet和Android手机进行交互,方法是在Spring的ApplicationContext.xml文件中增加下面的配置 (下转第102页)

信道中加入指定信噪比的零均值高斯白噪声, LMS 算法的均衡器抽头数为 11, 调整步长为 0.01, RLS 算法均衡器抽头数为 25, 遗忘因子为 0.98, 调整步长为 0.01。其中图 4(a) 图是脉冲信号经发送成形滤波器后输出的时域波形, 图 4(b) 是图 4(a) 信号的频谱图。从图 4(c) 和 (d) 可以看出 11 抽头 LMS 自适应均衡器的误差信号要大于 25 抽头 RLS 自适应均衡器的误差信号, RLS 算法自适应均衡器性能更优。从图 4(e) 和 (f) 可以看出光纤色散导致码间干扰的产生。从图 4(g) 和 (h) 可以看出 LMS 自适应均衡器和 RLS 自适应均衡器均能够改善信号输出, 有效消除码间干扰, 25 抽头 RLS 自适应均衡器比 11 抽头 LMS 自适应均衡器的均衡性能要好, 收敛速度更快, 但是以提高计算量为代价的。

4 结 论

LMS 算法和 RLS 算法都比较简单、运算高效性, 在各种条件下具有良好性能。对于高速光纤通信系统 LMS 算法和 RLS 算法同样能很好的对光纤信道进行均衡, 有效消除光纤色散和 PMD 引起的码间干扰。但 RLS 算法稍微复杂, 但收敛速度更快, 性能更好。所以, 基于这两种算法的自适应均衡器

都能很好满足对均衡速度要求较高的光纤通信系统。

参考文献:

- [1] 顾晓仪, 闻和. WDM 超长距离光传输技术[M]. 北京: 北京邮电大学出版社, 2006.
- [2] Buchali F, Bulow H, Baumert W, et al. Reduction of the chromatic dispersion penalty at 10Gbit/s by integrated electronic equalizers [C]//in Techn.Dig.OFC'00, Baltimore, Mar.7-10, ThS1 2000, 268-270.
- [3] Ingham J D, Penty R V, White I H, et al. Electronic equalization for length extension of $\times 2$ to $\times 3$ in 10Gbit/s multimode-fiber data communication links [J]. Electron.Lett., 2004, 40(22): 1473-1439.
- [4] Bulow H, Buchali F, Baumert W, et al. PMD mitigation at 10Gbit/s using linear and nonlinear integrated electronic equalizer circuits[J]. Electron.Lett., 2000, 36(2): 163-164.
- [5] 樊昌信, 张甫翊, 徐炳祥, 等. 通信原理 [M]. 5 版. 北京: 国防工业出版社, 2001.
- [6] 张贤达. 现代信号处理 [M]. 北京: 清华大学出版社, 2002.

(上接第 98 页)

即可。ApplicationContext.xml 文件主要配置如下:

```
<beans>
  <!-- 下面是手机终端通过 Spring 访问网站的必配内容 -->
  <bean id="CustomerServlet" class="com.bai.android.servlet.CustomerServlet">
    <property name="cusDao">
      <!-- 将 Spring 业务交给 Servlet 对象 -->
      <ref bean="customerServiceTarget"/>
    </property>
  </bean>
</beans>
```

这样服务器响应给手机终端的数据处理部分完毕, 一次交互过程完成。

3 结束语

Android 提供的 API 仅仅把从 Web 服务器中的数据响应到手机终端的 HttpServletResponse 对象中, 接着将该对象中的数据转化成字节数组, 由于响应过来的数据可能是各种不同的类型, 都包含在该字节数组中, 要将自己数组中的数据还原成服务器端原有的数据类型, 就要采用别的方法, 于是采用了 Sun 公司提供的 ByteArrayInputStream^[6]过滤流和 DataInputStream 流绑定的方法, 利用 DataInputStream 提供的各种 readXxx() 方法, 就可以把 Web 服务器端发过来的不同数据类型 Xxx 数据提取出来, 然后交给 Android 提供 View^[7]对象在界面上进行处理。这样就为 Android 手机访问 Web 服务器

提取不同类型的数据提供了技术构架^[8-9], 为 Android 和其他种类的手机开发人员提供了参考。

参考文献:

- [1] Burnette Ed. Android 基础教程[M]. 张波, 高朝勤, 杨选, 等译. 北京: 人民邮电出版社, 2009.
- [2] 彭晨阳. Java 实用系统开发指南[M]. 北京: 机械工业出版社, 2004.
- [3] 王向辉, 张国印, 陈洁. Android 应用程序开发[M]. 北京: 清华大学出版社, 2010.
- [4] 廖文奎. Struts 编程基础与实例精讲[M]. 北京: 中国电力出版社, 2006.
- [5] Deitel P J, Deitel H M. Java for programmers[M]. 张君施, 译. 北京: 电子工业出版社, 2010.
- [6] 罗会波. JSF 第一步[M]. 北京: 清华大学出版社, 2007.
- [7] 余志龙. Google Android SDK 开发范例大全 [M]. 北京: 人民邮电出版社, 2009.
- [8] 连晋平, 刘平. Web Service 技术在移动通信商务手机终端的应用[J]. 电子设计工程, 2008, 16(7): 64-65.
- LIAN Jin-ping, LIU Ping. Application of Web Service in mobile-communication-business cell phone[J]. Electronic Design Engineering, 2008, 16(7): 64-65.
- [9] 连晋平, 李梅, 刘平. 3G 移动商务系统的数据层次和状态分析[J]. 电子设计工程, 2010, 18(2): 132-133.
- LIAN Jin-ping, LI Mei, LIU Ping. Analysis of data levels&states of 3G mobile business affairs system[J]. Electronic Design Engineering, 2010, 18(2): 132-133.