

基于 JSON_RPC 异步通讯技术的 WEB 实现

王 晓

(青岛远洋船员学院现代教育技术中心, 山东 青岛 266071)

提 要:在 JSON 技术基础上,将 JSON_RPC 框架应用于 WEB 开发,把面向对象操作引入到 JavaScript 编程,改进了传统 AJAX 异步通讯技术,克服了 AJAX + JSON 技术存在的缺陷。通过 JavaScript 与 Java 对象间的数据转换以及远程方法调用,实现了客户端编程的简化处理,达到了“富服务器,瘦客户机”的目的。对于改善网页的局部刷新、解决客户端唯一性验证等问题提供了有效的解决方案。

关键词:异步通讯技术 远程方法调用 JSON_RPC 框架

中图分类号: TP391

文献标识码: A

异步通讯技术^[1] AJAX (Asynchronous JavaScript and XML) 为传统的 Web 架构提供一种异步数据传输机制,它相当于用户和服务器之间的一个中间层,其功能使用户请求与服务器响应异步化。但 AJAX 异步通讯技术与面向对象的 WEB 开发思想不一致。面向对象 WEB 开发要求服务器程序、客户端程序 (甚至是网页程序) 都要面向对象进行操作,传统的 AJAX 异步通讯技术很难做到这一点。本文在 WEB 开发中引入了基于 JSON_RPC 异步通讯技术,尽可能多地将访问 WEB 服务器与访问数据服务器的操作放在服务器端执行,只在客户端提供调用 WEB 操作的接口,不仅实现了 WEB 开发的异步通信,而且将面向对象操作应用于客户端程序开发。

1 JSON 简介

JSON (JavaScript Object Notation) 是一种轻量级的数据交换格式^[2]。它基于 JavaScript,采用完全独立于语言的文本格式来表示数据,在 AJAX 中常用来与服务器端交换数据。

1.1 JSON 构建的结构

1) “名称/值”对的集合 (A collection of name/value pairs)。在不同的语言中,它被理解为对象

(Object), 记录 (Record), 结构 (Struct), 字典 (Dictionary), 哈希表 (Hash table), 有键列表 (Keyed list), 或者关联数组 (Associative array)。

2) 值的有序列表 (An ordered list of values)。在大部分语言中,它被理解为数组 (Array)。这些都是常见的数据结构,大部分现代计算机语言都以某种形式支持它们,这使得一种数据格式在同样基于这些结构的编程语言之间的交换成为可能。在面向对象的 JAVA EE WEB 开发中,JSON 可以作为对象 (Object) 进行处理,从而将服务器端定义的 JAVA 对象方便地引入客户端,从而通过引入的对象非常方便地调用其定义的方法,访问其定义的属性。

1.2 JSON 具有的对象形式

JSON 表示的对象是一个无序的“‘名称/值’对”集合。一个对象以“{”(左括号)开始,“}”(右括号)结束。每个“名称”后面跟一个“:”(冒号);“‘名称/值对’”之间使用“,”(逗号)分割。

1.3 注册对象的 JSON 表示

有一网络注册数据包括:用户名、密码、电子邮件等。用 JSON 表示其部分数据结构的 BNF 定义如下:

收稿日期: 2009 - 11 - 19

作者简介: 王晓 (1971 -), 男, 硕士, 讲师

```
{
  "name": "张三",
  "password": "*****",
  "email": "zhangsan@126.com"
}
```

以上结构表示一个 JSON 对象,其属性分别是“name”、“password”、“email”,其对应的值分别为“张三”、“*****”和“zhangsan@126.com”。从以上结构来看 JSON 非常适合于客户端与服务器进行数据交换处理,合理地使用 JSON,不仅可以提高 AJAX 的通讯效率,而且可以提高 AJAX 应用的开发效率。

2 AJAX + JSON 技术与 JSON_RPC 的比较

传统的异步通讯技术习惯于采用 AJAX + JSON 形式^[3]进行编程,这需要在 html 文件或者 js 文件中,通过定义 JavaScript 函数创建 XMLHttpRequest 对象,通过 XMLHttpRequest 对象实现客户端与服务器的信息交换。创建 XMLHttpRequest 对象的部分 JavaScript 代码如下:

```
<script language="javascript">
  //定义一个将指向 XMLHttpRequest 对象的
  变量
  var xmlhttp;
  //定义一个函数用于创建 XMLHttpRequest
  对象
  function createXMLHttpRequest() {
    if (window.ActiveXObject) { xmlhttp =
    new ActiveXObject("Microsoft.XMLHTTP");
  }
    else if (window.XMLHttpRequest) {
      xmlhttp = new XMLHttpRequest();
    }
  }
  ...../省略 XMLHttpRequest 对象获得网页数
  据与服务器传送数据代码
</script>
```

同时,负责服务器与客户端进行信息交互的程序必须满足以下条件:1)要扩展 HttpServlet 类;2)服务器与客户端所有交互操作都需要放到 doGet()与 doPost()方法中。服务器端用于异步通讯部分 JAVA 代码如下:

```
public class JsonUser extends HttpServlet {
  public void doGet(HttpServletRequest
```

```
request, HttpServletResponse response)
  throws ServletException, IOException {
  doPost(request, response);
}

public void doPost(HttpServletRequest
request, HttpServletResponse response)
  throws ServletException, IOException {
  //读取客户端提交的 JSON 数据
  ...../省略代码
}
  //建立一个服务端的 JSONObject 对象解析
  客户端的 JSON 数据
  ...../省略代码
  //从 JSONObject 对象中取出各项数据进行
  处理
  ...../省略代码
}
```

从以上分析可以看出 AJAX + JSON 技术存在两方面弱点:一是由于采用 AJAX + JSON 框架后,很多 View 和 Control 的代码需要转移到前台用 JavaScript 实现,使客户端的 JavaScript 编程过于繁杂,与“瘦客户机”的 WEB 开发理念相悖;二是服务器端程序必须使用 doGet()与 doPost()方法来处理与客户端的数据交互,必须使用 HttpServletRequest 对象的参数 request 和 HttpServletResponse 对象的参数 response 在服务器和客户端传值,这导致程序编写的依赖性强,灵活性差,并且参数定义繁琐,不便拼写记忆,令编程者厌烦。本文在 JSON 技术的基础上,通过引入 JSON_RPC 框架,对 AJAX 异步通讯技术进行了改进,克服了 AJAX + JSON 技术存在的弱点:1)通过在客户端文件中使用 JsonRpcClient 对象,直接访问服务器的 JAVA 程序,能够达到“富服务器,瘦客户机”的目的;2)服务器上的类对象以 JavaBean 的形式定义,不需要扩展 HttpServlet 类,不必直接使用 doGet()与 doPost()方法和 request 与 response 参数传值,因而编程灵活、记忆简单,为程序员开发留有充分的发挥余地。

3 JSON_RPC 原理与 JAVA WEB 应用中的改进

JSON_RPC_Java (简称 JSON_RPC)是一个用 JAVA 来实现动态 JSON_RPC 的框架^[2],是基于 JSON 架构的对服务器对象的远程调用。它由两个对用户可视化的组件 (JSONRPCBridge, JSONRPC-

PCServlet)和一个内置的轻量级客户端 (JSON - RPC JavaScript)组成。JSONRPCBridge 和 JSONRPCServlet 二者的协调实现了服务器端对请求对象的处理,并将处理结果响应给用户;轻量级客户端 JSON - RPC JavaScript 可以透明地在客户端 JavaScript 中调用服务器端的 JAVA 程序,实现客户端的面向对象操作。

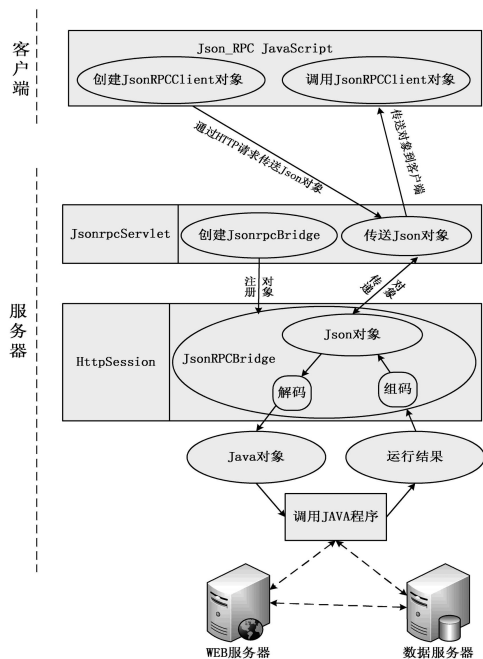


图 1 JSON - RPC工作原理图

3.1 原理

客户端 JSON_RPC JavaScript 通过 new JsonRPCClient("JSON_RPC") 语句创建 Json 对象,并将 Json 对象通过 HTTP 请求传递给 JSONRPCServlet。JSONRPCServlet 首先根据客户端请求,在 HttpSession 中为每一个用户创建一个 JSONRPCBridge 实例,并为该实例以属性名 "JSONRPCBridge" 在 session (会话) 中注册,然后在客户端与服务器之间充当传送器角色,将 HTTP 传输的 JSON_RPC 请求交给 JSONRPCBridge 实例处理;JSONRPCBridge 接收服务器 (JSONRPCServlet) 传来的 JSON_RPC 请求,把 JSON 对象解码 (Unmarshalling) 为 Java 对象,通过 Java 对象调用其他 JAVA 程序,访问 WEB 服务器和数据服务器,然后将 Java 对象结果交给 JSONRPCBridge 组码 (Marshall) 为 JSON 对象 (调用序列化类 Serializer 处理完成),通过 JSONRPCServlet 将 JSON 对象传给客户端。工作原理如图 1。

3.2 JSON_RPC 技术改进

与 AJAX + JSON 技术相比,采用 JSON_RPC 技术需要从服务器和客户端两个方面进行编程改进 (以网上注册为例)。

3.2.1 服务器程序

1) 异步通讯业务中引入 JavaBean。JavaBean 是一个实现了某个功能且供重复使用的 JAVA 类。JSP 页面中使用 JavaBean 后,可以实现 HTML 与 JAVA 代码的分离,使 JSP 更易于开发和维护。注册对象 JavaBean 要满足以下四个条件: (1) JavaBean 对象中,属性要包含所有注册字段,即注册字段集合是 JavaBean 中属性集合的一个子集,从字段集到对象属性集是一个一对一的单一映射关系 (图 2); (2) JavaBean 是一个 public 的类; (3) JavaBean 有一个不带参数的构造方法; (4) JavaBean 中的每一个属性通过 getXXX () 和 setXXX () 进行操作,其中 XXX 代表属性名。

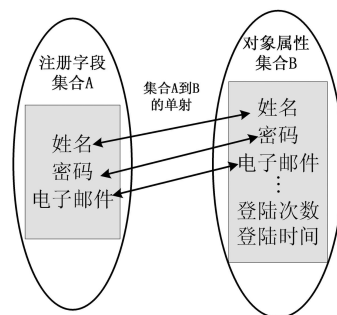


图 2 字段集到属性集的映射关系

2) 定义对象关系映射 ORM (Object Relational Mapping)。ORM 是通过使用描述对象属性与数据库字段之间对应关系,来做到将 JAVA 程序中的对象自动持久化到关系数据库中,从而达到对数据库的操作转化为对 JAVA 对象的操作。定义对象关系映射 ORM 要满足以下两个条件: (1) 对象属性集合与数据库中数据表的字段集合是一一对应的双射关系; (2) 对象类名与数据库中的表名要一一对应,如: < class name = "RegMember" table = "member_table" >, 其中, RegMember 为注册对象类名, member_table 为数据库中存放注册成员的数据表名。对象属性与数据库表的对应关系如图 3 所示。

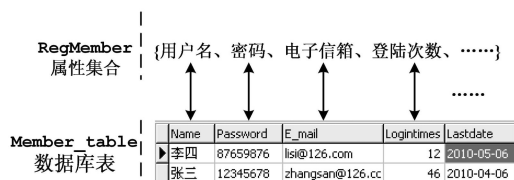


图 3 对象属性与数据库表的对应关系图

3)定义 JAVA 对象的数据库操作 DAO (Data Access Object)。需要完成以下三步:(1)扩展 HibernateDaoSupport功能,获得操作数据库的会话;(2)定义 JAVA 对象的 CRUD操作 创建(CREATE)、读取(READ)、更新(UPDATE)、删除(DELETE),程序通过对 JAVA 对象的操作,达到操作数据库的目的,实现关系型数据库的面向对象操作;(3)定义按照属性获得对象的方法,从而达到根据对象的属性值,获得数据库中对应的记录,实现关系型数据库查询的面向对象操作。JAVA 数据访问对象实例如图 4。

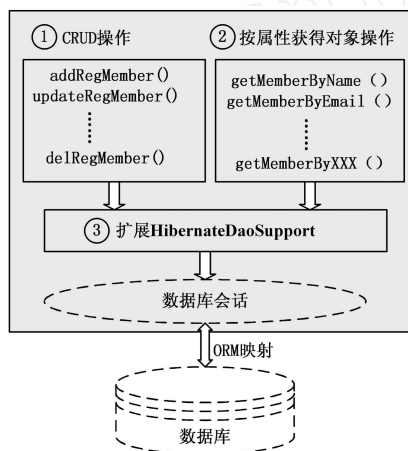


图 4 JAVA 数据访问对象实例

4)定义 AjaxAction。Action 是一个专门负责获得客户端 Form 表单数据并处理业务逻辑的 JAVA 类。AjaxAction 用于实现服务器与客户端异步通讯时,信息交换的业务逻辑处理,实现 JSON 对象与 JAVA 对象的交互;通过调用 JAVA 对象的数据库操作 DAO,达到按字段获取数据对象的目的。AjaxAction 需要实现以下三个方面的功能:(1)根据 Form 表单对应的 JSON 对象,创建对应用户对象实例;(2)通过创建的用户对象实例,调用其对应的 DAO 类中的操作;(3)将操作结果通过 JSON_RPC 传给客户端页面。

在 Struts2.0 中 AjaxAction 与外界交互,实现数值传递主要有两种方式^[4]:一种是通过字段驱动方式(Field-driven)传值,另一种是通过模型驱动方式(Model-driven)传值。所谓的字段驱动方式就是通过字段进行数据传递。在 Struts2.0 中可以直接在 AjaxAction 里定义各种 JAVA 基本类型的字段,使这些字段与表单数据相对应,并利用这些字段进行数据传递。所谓模型驱动是指通过 JavaBean 模型进行数据传递。采用这种方式,Java-

aBean 所封装的属性与表单数据的属性一一对应,JavaBean 将成为数据传递的载体。使用模型驱动方式,AjaxAction 类通过 getObject()方法来获取模型,其中“Object”代表具体的模型对象。

3.2.2 客户端程序

客户端程序引入 JsonRpcClient 对象代替 XMLHttpRequest 对象。为达到这一目的,需要进行如下操作:

- 1)将 jsonrpc.js 文件导入 WebRoot 的某个位置;
- 2)将 jsonrpc.jar 文件包导入到 WEB-INF/lib 下;

3)在系统配置文件 web.xml 中注册 JSONRPCServlet:

```
<servlet>
  <servlet-name>
    JSONRPCServlet
  </servlet-name>
  <servlet-class>
    com.metaparadigm.jsonrpc.JSONRPCServlet
  </servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>
    JSONRPCServlet
  </servlet-name>
  <url-pattern>
    /JSON-RPC
  </url-pattern>
</servlet-mapping>
```

4)在需要进行异步通讯的 JSP 文件注册 JSONRPCBridge

```
<jsp:useBean id="JSONRPCBridge"
  scope="session"
  class="com.metaparadigm.jsonrpc.JSONRPCBridge"/>
```

5)将 jsonrpc.js 导入 JSP 文件

```
<script type="text/javascript"
  src="jsonrpc.js"></script>
```

6)将服务器端的 JavaBean 对象放置到要使用 JSON_RPC 的 JSP 文件中;

7)将服务器端的 JavaBean 对象注册到 JSONRPCBridge

JSONRPCBridge registerObject (" JavaBeanObject", ajax);

8)在 JavaScript脚本中创建一个 JSONRpcClient对象

jsonpc = new JSONRpcClient (" JSON _RPC");

9)利用创建的 JSONRpcClient对象 jsonpc,在客户端 JavaScript程序中直接调用服务器端的 JAVA 程序。

4 结论与局限性分析

JSON_RPC为传统的 Web架构提供了一种新颖的异步数据传输机制,通过使用 JSON_RPC架构对服务器和客户端程序进行改进,将 JSON_RPC异步通讯技术运用于 Java EE WEB 开发,极大提高了网络响应速度、运行效率和用户交互性,对于使用异步通信技术解决客户端唯一性验证等问题提供了有效的解决方案,对于要求局部页面刷新的网络编程非常有用;它将面向对象的操作引入到 JavaScript编程,实现了 JavaScript与 Java对象间的数据转换与远程方法调用,达到了客户端编程简化处理的目的;通过将负责异步通信操作的 JAVA 程序放在服务器上运行,客户端通过 JSON-RpcClient对象直接访问服务器的 JAVA 程序,而不是将程序下载到客户端运行,达到“富服务器,瘦客户机”的目的。

由于异步通信技术严重地依赖于客户端技术(如 Javascript等),而 Web客户端使用的浏览器各不相同,各种浏览器提供的客户端 API函数也存在一定的差别,即使同一厂商提供的浏览器程序也存在着版本的差别。所以在使用 JSON_RPC异步通讯技术进行 Web程序开发时,一定要考虑到对主流浏览器兼容的问题,需要针对不同的浏览器提供的 API函数编写不同的程序代码。同时,对于低版本的或者不支持客户端技术的浏览器,还需另外编写不能应用 JSON_RPC异步通讯技术的代码,以使得 Web程序能够在这些浏览器上运行。

参考文献:

- [1] Jesse James Garrett Ajax: A New Approach to Web Applications [EB /OL]. <http://adaptive-path.com/publications/essays/archives/000385.php>, 2005.
- [2] 李德贤,陆歌皓等. JSON_RPC协议分析、扩展及其应用 [J]. 中国科技论文在线, 2008, 3 (2): 125 - 130.
- [3] 刘斌. 精通 Java Web整合开发 [M]. 北京:电子工业出版社, 2007. 11
- [4] 三杨科技 开发者突击: Struts 2核心技术与 JAVA EE框架整合开发实战 [M]. 北京:电子工业出版社, 2008. 10

JSON_RPC A synchronous Communication Technology based on the WEB

WANG Xiao

(Modern Education Technology Center, Qingdao Ocean Shipping Mariners College,
Qingdao 266071, China)

Abstract: Based on JSON technology, the JSON_RPC framework is applied to WEB development, by which object - oriented operation is introduced into JavaScript programming and which improve the traditional Asynchronous JavaScript and XML technologies to overcome AJAX & JSON technical defects. Through the data conversion between the JavaScript and the Java objects and remote method invocation, the client program can be simplified, and the aim of " rich server, thin client" is reached meanwhile, an effective solution is provided, which not only improves the local refreshment but also give rise to the only client - side validation.

Keywords: A synchronous JavaScript and XML, remote method invocation, JSON_RPC framework