

基于两种 Web 开发模式构建 AJAX 应用的研究

宋瑜辉, 马光思

(西安建筑科技大学 信息与控制工程学院, 陕西 西安 710055)

摘要: 为了提高 Web 应用的访问效率, 针对 J2EE 平台下两种基于 AJAX 的框架 JSON-RPC-Java 与 DWR, 分别给出了二者与其它框架集成的组成结构、具体整合过程和应用实现的关键技术。结合应用研究与工程实践, 分析总结了两种基于 AJAX 开发模式的不同技术支持, 在多个层面为 Web 应用开发提供了更大的便捷和支持。研究工作对开发方案选型和工程应用都有很好的参考价值。

关键词: JSON-RPC-Java; DWR; 框架集成; 技术支持

中图分类号: TP311.5

文献标识码: A

文章编号: 1673-629X(2009)05-0086-03

Research of Constructing AJAX Application Based on Two Web Development Models

SONG Yu-hui, MA Guang-si

(School of Info. and Control Eng., Xi'an Univ. of Architecture and Tech., Xi'an 710055, China)

Abstract: Aiming at JSON-RPC-Java and DWR, which are two kinds of frames based on AJAX on J2EE platform, present separately the composition structure, the key technologies of the concrete integrating process with other frameworks and the application implementation. Associated with application research and project practice, analyzed and summarized the different technology support between the two development models based on AJAX, which can give more convenient support in several layers. The research works has the very good reference role to the development scheme selection and the project application.

Key words: JSON-RPC-Java; DWR; framework integration; technology support

0 引言

传统 Web 应用基于同步交互方式, 导致了应用的访问速度低效。AJAX 技术的出现从提高网络带宽、分担服务器负担、节省消费资源、异步交互和局部刷新等多个方面把 Web 应用带进了 2.0 时代。

AJAX 技术通过 XMLHttpRequest 对象实现 Web 应用的异步交互, 但使用 XMLHttpRequest 创建异步请求的过程较为烦琐。因为, 要编写复杂的 JavaScript 代码, 要考虑如何定位 AJAX 请求到适当的服务, 以及处理跨浏览器 DOM 实现兼容性等。对此, 一些组织发布了基于浏览器和基于服务器的软件框架, 用以支持、简化 AJAX 应用的开发。

(1) 基于浏览器端的 AJAX 框架, 实现原理较为简

单, 只需封装 JavaScript, 保留用户设置部分。开发 AJAX 应用相当于调用 JavaScript 组件对象, 并给这些组件设置属性值。

目前常用的浏览器端的框架有: Bindows, Backbone, Dojo, Open Rico 等^[1,2]。

(2) 服务器端框架通常有两种工作方式:

① HTML/JavaScript 生成。由服务器端生成 HTML 和 JavaScript 代码, 并传递给浏览器端直接运行;

② 远程调用。客户端 JavaScript 调用直接被定位到服务器端的相应函数, 并返回数据给客户端 JavaScript 的回调函数, 或者由客户端 JavaScript 请求服务器端数据信息。

采用 Java 平台的 AJAX 框架多为开源项目, 当前流行的服务器端框架中, 以 DWR^[3] 和 JSON-RPC-Java 应用最为广泛。文中的研究工作主要针对这两种框架的应用实现。

实际应用开发中, DWR 与 JSON-RPC-Java 均需与其它软件框架集成, 才能互相协作, 发挥各自的重

收稿日期: 2008-08-31

基金项目: 陕西省自然科学基金(2005F38), 陕西省教育科研基金(07JK306)

作者简介: 宋瑜辉(1983-), 女, 陕西西安人, 硕士研究生, 主要研究领域为计算机理论; 马光思, 教授, 主要研究领域为计算理论、信息安全、Web 技术。

要作用。

1 集成 Spring + Hibernate 与 DWR

参见图 1^[4],集成框架将 Web 应用分为 4 层,具有 MVC 模式的优点。

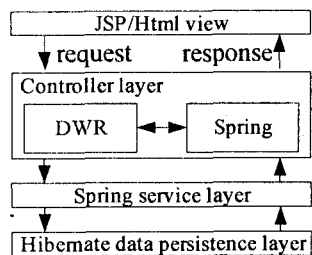


图 1 集成 Spring + Hibernate 与 DWR

(1) Hibernate 框架自然位于持久层,发挥 ORM 的作用,负责数据持久层的管理,映射数据表与 Java 对象。这样, Spring 服务层就可以通过 Java 对象管理底层数据^[5,6];

(2) Spring 服务层实现业务逻辑处理功能,用作中间层的 IoC 容器,以松耦合的方式组织各组件,基于 Spring 的 DAO 实现具体的 DAO 对象,定义功能模块。它向下直接与 Hibernate 数据持久层交互,向上提供功能接口,可以由控制器直接调用,并返回最终处理结果^[4,7];

(3) DWR + Spring 作为控制器,实现请求处理及转发功能。DWR 在这层用作 Ajax 引擎,负责处理功能调用请求; Spring 控制器负责处理页面重定向的请求。具体集成过程见 4.1 节;

(4) 视图层用 JSP/Html 实现与用户的交互。

集成框架通过 DWR 直接访问 Spring 容器中的业务逻辑组件,利用 Spring 容器的功能提高浏览器访问效率,优化 Web 开发模式内部结构,提高可维护性。

2 集成 Hibernate 与 JSON - RPC - Java

参见图 2,集成框架将 Web 应用分为 4 层,保持 MVC 模式的优点。

(1) Hibernate 负责数据持久层的管理,直接与数据库交互,映射数据表与 Java 对象。使用 DAO 模式访问持久层^[4,7];

(2) 业务逻辑组件用来封装业务逻辑方法,而这些业务逻辑方法将被公开给远程 JavaScript 客户端。业务逻辑组件(服务器处理类)基于 DAO 组件实现,并通过 JSONRPCBridge 公开成一个 JavaScript 对象,从而可以在客户端调用该对象的方法;

(3) 因为 JSON - RPC - Java 暂时还不能直接调用 Spring 容器中 Bean,这可能会限制其使用。但 JSON -

RPC - Java 是 J2EE 领域的一个优秀框架,发挥 RPC 调用的作用。与 DWR 类似,JSON - RPC - Java 也能将普通的 Java 对象的方法直接公开给 JavaScript 客户端,允许从 JavaScript 客户端直接调用。公开方法是通过将 Java 对象注册成客户端的 JavaScript 对象来实现的,只是注册过程略有区别。一旦将远程对象转换成对应的 JavaScript 对象,便可以通过回调函数调用服务器的 Java 方法。JSON - RPC - Java 公开远程 Java 对象主要依赖于 JSONRPCBridge 和 JSONRPCServlet。其中,具有特殊拦截器功能的 JSONRPCServlet 负责拦截用户请求,JSONRPCBridge 是客户端 JavaScript 对象和 Java 对象之间的桥梁,负责将拦截到的用户请求转发给用户处理类并将处理结果返回给 JSONRPCServlet;

(4) 视图层用 JSP 实现与用户的交互。

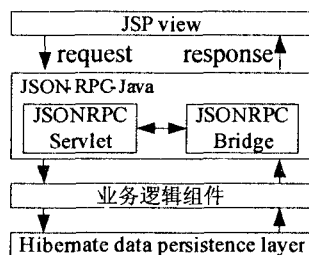


图 2 集成 Hibernate 与 JSON - RPC - Java

3 框架集成与构建应用实例

通过书目管理实例,分别介绍这两种模式的应用,其中只涉及读取和显示功能。环境为 Eclipse + Tomcat + MySQL。

3.1 集成 Spring + Hibernate + DWR 及应用示例

(1) 开发环境配置。

STEP1 在 web.xml^[8]中配置 Spring 和 Hibernate 框架。

STEP2 在 web.xml 文件中增加一个名字为 dwr - invoker 的 servlet,其 class 是 uk. ltd. getahead. dwr. DWRServlet。所映射的 url - pattern 是 /songdwr / *, 表示 DWR 所公开的 JavaScript 对象都位于 songdwr URL 下。在客户端 JavaScript 代码中必须导入位于该 URL 的 JavaScript 代码库。

STEP3 在 /WEB - INF 目录下创建 application - Context.xml 文件,并配置以下两个 JavaBean: 一个是 Chat - Service, id 为 chatService; 另一个是 ChatDwr, id 为 chatDwr,其中 chatDwr 中要引用 chatService 作为属性。

STEP4 在 /WEB - INF 目录下创建 dwr.xml^[7,8]文件,并添加如下代码:

```

<create creator="spring" javascript="bm">
<!-- 定义将 Spring 容器中的 bookManager Bean 创建
成 JavaScript 对象 -->
<param name="beanName" value="bookManager"/>
<!-- 下面每个 include 元素定义了一个可被公开的方法 -->
<include method="createComment"/>
<include method="createBook"/>
<include method="getAllBookByPage"/>
<include method="getCommentsByBookAndPage"/>
<include method="getBook"/>
<include method="getNewestBook"/>
</create>
<!-- 下面定义了两个 JavaBean 可被 DWR 通过 bean
转换器转换成 JavaScript 对象 -->
<convert converter="bean"
match="org.song.vo.CommentBean"/>
<convert converter="bean"
match="org.song.vo.BookBean"/>

```

完成以上的配置,即可在 Javascript 代码中直接调用 bm 对象的方法——blogManager 的方法。

(2) 应用实现。

编写程序代码,需要创建以下文件:

- ① book.html:用于页面信息的显示。
- ② BookManager.java 文件:提供功能接口,调用底层服务。
- ③ BookManagerImpl.java 文件:实现业务逻辑。
- ④ Book.java 文件:实现 O/R 映射,通过 Hibernate 映射为存储书目信息的数据表。

STEP1 在 book.html 中要添加 /songdwr/interface/bm.js、/songdwr/engine.js、/songdwr/util.js 这要求在客户端 JavaScript 代码中导入上面的 3 个 JavaScript 函数库。在 Javascript 中可以通过调用 bm 操作 DWR 对象 bookManager。

STEP2 在 bm.js 中通过调用 bm.getNewestBook(bookCb)方法得到最新发布的书目;通过调用 bm.getBook(id,bookCb)方法得到指定 ID 的 Book 内容。

至此,就可以将书目信息显示在浏览器中。整个过程并没有对 JSP 页面进行刷新,只是通过 DWR 提取数据,然后对数据进行处理,完成异步交互过程。重定向页面的请求和不适合异步处理的请求则发给 Spring 控制器处理。

3.2 集成 Hibernate 与 JSON-RPC-Java 及应用示例

(1) 开发环境配置:

STEP1 在 web.xml^[8]中配置 Hibernate 框架。

STEP2 在 web.xml 文件中增加一个名字为 json-

Servlet 的 servlet,其 class 是 com.metaparadigm.jsonrpc.JSONRPCServlet,所映射的 url-pattern 是 /myjson。一旦完成了上面的映射,便表示所有向 /myjson 地址的请求都将发给 JSONRPCServlet 处理。因此,JSON-RPC-Java 框架中的另一个核心类 JSONRpcClient 应以此 URL 创建实例。

STEP3 完成 JSONRpcClient 的初始化,在应用的 JavaScript 代码库中增加如下函数:

```
jsonrpc = new JSONRpcClient("myjson");
```

(2) 应用实现。

编写程序代码,需要创建以下文件:

- ① book.html:用于显示页面信息。
- ② BookManager.java 文件:提供功能接口,调用底层服务。
- ③ BookManagerImpl.java 文件:实现业务逻辑。
- ④ Book.java 文件:实现 O/R 映射,通过 Hibernate 映射为存储书目信息的数据表。

STEP1 在 book.jsp 中要添加如下代码片段:

```

<!-- 初始化 JSONRPCBridge 对象 -->
<jsp:useBean id="JSONRPCBridge" scope="session" class
="com.metaparadigm.jsonrpc.JSONRPCBridge" />
<!-- 初始化 BookManager 对象 -->
<jsp:useBean id="as" scope="session"
class="org.song.service.BookManager" />
<% //注册 BookManager 对象,将其转换为一个客户端
JavaScript 的一个 as 对象
JSONRPCBridge.registerObject("bm", bm);
%>

```

将远程 Java 对象注册在该对象下。BookManager 对象将被公开成一个 JavaScript 对象,这个对象作为 JSONRpcClient 对象的一个属性引用。

STEP2 在 bm.js 中通过调用 bm.getNewestBook(bookCb)方法得到最新发布的书目;通过调用 bm.getBook(id,bookCb)方法得到指定 ID 的 Book 内容。

至此,就可以将书目信息显示在浏览器中。整个过程并没有对 JSP 页面进行刷新,只是通过 JSON-RPC-Java 提取数据,然后对数据进行处理,完成了异步交互过程。

4 相关技术支持的比较分析

DWR 和 JSON-RPC-Java 都是 J2EE 领域的 AJAX 框架,专注于为 Java Web 应用增加 AJAX 能力,共同之处在于都允许 JavaScript 代码直接调用远程 Java 方法,但它们之间在应用开发的技术支持方面,还存在着很大差异,参见表 1。

(下转第 92 页)

重复次数较少时,压缩效果也不好。

此外对大小为 1M 的中文数据也进行了试验,发现效果没有英文的好,这是因为英文的重复和规律行比中文要好,而中文的信源远多于英文。

同时在实验的后期选用不同的实验数据与 PKZIP 进行了比较。实验结果见表 2。

表 2 不同材料的实验结果

| 文件名称 | 原始文件大小 | PKZIP 文件大小 | PKZIP Bits/Byte | Bwt 文件大小 | BwtBits/Byte |
|------|----------|------------|-----------------|----------|--------------|
| 书本 | 610, 856 | 209, 061 | 2.74 | 186, 592 | 2.44 |
| 文章 | 38, 105 | 13, 331 | 2.80 | 13, 159 | 2.76 |
| 新闻 | 377, 109 | 146, 010 | 3.10 | 134, 174 | 2.85 |

总之 BWT 转换后字符内聚性在很大程度上影响最终压缩的比率。只有转换后表现出良好内聚性的字符才能得到较好的压缩比率。

5 结束语

BWT 变换及逆变换的作用虽然不是直接的压缩,却给其他文本压缩方法提供了很好的基础。BWT 独特的可逆轮换方式使得原本没有规律的随机字符串呈现出局部的字符内聚性质,即使部分字符在转换中以扎堆的形式连续出现,这一重要特性方便了其它编码,节省了其他编码对原始字符串的处理过程,同时又能达到更高效的压缩比率改变了文本的固有属性,在一

段程度上拓展了一些原有的压缩算法的应用范围。同时,BWT 算法本身由于包含了大量的排序操作,从而造成了较高的时间复杂度,相信现在和不久的将来会有不少学者在对此进行研究。同时基于 BWT 的多媒体数据压缩是需要进一步研究的方向。

参考文献:

[1] 黄 锐,唐继勇.文本类型气象雷达原始回波数据的无损压缩[J].通信与信息技术,2008(1):33-34.

[2] 祝 庚,钟嘉鸣.数据压缩与解码技术探讨[J].湘南学院学报,2002(5):20-22.

[3] Bunton S. On-Line Stochastic Processes in Data Compression [D]. USA:Dept. of Computer Science and Engineering, University of Washington, 1996.

[4] Burrows M, Wheeler D J. A Block-sorting Lossless Data Compression Algorithm [R]. SRC Research Report 124 [s. l.]:Digital Systems Research Center, 1998.

[5] 白跃彬,卢华斌,韩庆绵,等.图像压缩技术及其进展[J].微机发展(现更名:计算机技术与发展),1996,16(4):97-98.

[6] Cleary J G, Teahan W J. Unbounded Length Contexts for PPM[J]. Thev Computer Journal, 1993, 36(5):32-36.

[7] Cormack G V, Horspool R N. Data Compressing Using Dynamic Markov Modeling[J]. Computer Journal, 1987, 30(6):541-550.

(上接第 88 页)

表 1 DWR 和 JSON-RPC-Java 的特点比较

| 应用开发的技术支持 | DWR | JSON-RPC-Java |
|-----------------|--|--|
| 后台服务的注册和访问控制方式 | 使用 dwr.xml 定义需要公开的服务,在前台 javascript 中用 <script src="xxx"> 把服务引进到客户端;服务的周期管理可以在 XML 中配置 | 默认在 Session 的 Bridge 里注册要使用的 Java 对象,这种方式有很多限制,使用时需要注册 JSP 或 Servlet 程序;服务的周期管理较麻烦 |
| 客户端页面形式 | HTML 页面、JSP 页面 | JSP 页面 |
| 与原有 J2EE 框架整合程度 | 能与 Spring 和 Struts 等框架无缝整合 | 不能与 Spring 和 Struts 等框架的整合应用 |
| 调试控制台 | 可在控制台调试注册远程 javascript 对象的详细信息 | 不提供调试控制台 |
| 支持反向 AJAX 程度 | 支持 | 不支持 |
| 其它 | 提供 DWRUtils 工具类简化 DOM 操作 | 不提供任何简化 DOW 操作工具 |

5 结束语

利用 AJAX 框架,简化 AJAX 应用开发,构建高效 Web 应用已是热点研究课题。文中结合工程实际,研究了基于 AJAX 技术构建 Web 应用的两种开发模式,通过具体实例分析了相关框架集成和应用实现的技术

要点。所实现的功能已取得了满意的应用效果和用户好评。

参考文献:

[1] 李 刚.基于 J2EE 的 Ajax 宝典[M].北京:电子工业出版社,2007.

[2] Garrett J J. Ajax: A New Approach to Web Applications[EB/OL]. 2005-02-18. <http://www.adaptivepath.com/publications/essays/archives/000385.php>.

[3] DWR:Easy AJAX for JAVA[EB/OL]. 2007-11-20. <http://getahead.org/dwr/overview/dwr>.

[4] 吴学义.基于 AJAX 的 B/S 架构及应用[J].吉林大学学报, 2007,3(13):314-318.

[5] 李 扬,马光思.扩展与整合 Web 应用框架的研究与实践[J].计算机工程,2006,32(10):144-146.

[6] 蒋 伟,马光思.Spring 与其他框架整合及流程分析[J].计算机工程,2007,33(14):79-81.

[7] 纪 颖,马光思.使用 DAO 和业务代理联合模式整合 Web 应用框架[J].计算机技术与发展,2006,16(11):19-21.

[8] Dynamic Html and XML:The XMLHttpRequest Object[EB/OL]. 2005-06-24. <http://developer.apple.com/internet/webcontent/xmlhttpreq.html>.