

RESTful Web Services 在信息系统中的应用^①

王 非, 蔡 勇, 贺志军

(北京师范大学珠海分校 信息技术学院, 珠海 519085)

摘 要: RESTful Web Services 是一个轻量级的架构, 其中包含的重要概念有资源的可寻址性、连通性、无状态、统一接口等. 通过低耦合和其他分布式组件, 使其具有良好的交互性和可伸缩性. 由于无需引入复杂的 SOAP 协议, 比起 RPC 风格的 Web Services, 很容易使用并且获得了广泛的应用. 研究了在一个中医药信息系统中如何使用 Restlet 框架构建 RESTful Web Services.

关键词: 表述性状态转移; 网络服务; 资源; 统一接口; 简单对象访问协议

RESTful Web Services Applied in Information System

WANG Fei, CAI Yong, HE Zhi-Jun

(College of Information Technology, Beijing Normal University, Zhuhai 519085, China)

Abstract: The RESTful Web Service is a lightweight framework which includes some important ideas such as addressability of resources, statelessness, connectedness, and uniform interface and has good interaction and elasticity by its loose coupling and other distributed components. Since there is no need to introduce the complex SOAP, it is easier to use, easier to combine than RPC-style services and has gained wide spread acceptance across the web. In this paper, it is researched how to use Restlet framework to build RESTful Web Services in a Chinese medicine information system.

Key words: REST; Web services; resource; uniform interface; SOAP

1 引言

在行业及企业的信息化过程中, 数据来源越来越广泛同时信息资源利用也极不平衡, 信息资源基本处于各自为政的局面(比如中医药行业). 鉴于体制机制的原因, 数据资源管理部门由各部门自行编目存储与管理. 虽然很多数据资源管理部门都提供了数据检索系统, 但这些数据检索系统很难实现相互之间的信息共享与协同访问, 形成了一个信息孤岛. 因此, 开展分布式的数据检索与共享研究迫在眉睫. RESTful Web Services 能够很好的实现跨平台、跨 Internet 接口调用又充分利用了现有 HTTP 协议, 是一种更适合于分布式应用的解决方案.

目前 Web Services 主要有两种风格, 一种是 XML-RPC 风格, 它以 SOAP 协议为信息传递的基础, 借助

HTTP 协议的传输功能, 采用面向动作的设计思路, 将所要交互的方法和消息包装成 SOAP 协议后, 放入 HTTP 的协议中传输, 以此来完成信息的传递和交互, 但是复杂的嵌套协议在给开发设计带来了不必要的麻烦的同时, 自定义的方法, 也使得不同系统间的交互产生阻碍. 也使得 HTTP 协议成为一种用于传输庞大 SOAP 负载的协议, 失去了 HTTP 设计的本意^[1]. 另一种风格是具有表述性状态转移风格的 Web Services 称为 RESTful Web Services, 是指使用 HTTP 协议的语法和语义将其功能完全作为一组 URI 可寻址资源来向服务消费者提供的 Web Services 架构方式. 它可以通过 Web 本身具有的松耦合特性以及其他分布式基础设施来实现具有很好交互功能和可伸缩性的 Web 服务. REST 即表述性状态转移 (Representational State Transfer)^{[2][3]}, 这是一术语是 2000

① 收稿时间:2012-07-26;收到修改稿时间:2012-09-17

年 Roy Thomas Fielding 在他的博士论文中提出的, 作为一种新的分布式超媒体系统设计架构改变了需要在 HTTP 中附加庞大负载的现状, 以面向资源的设计理念代替了传统的面向动作的设计理念. REST 风格架构让 HTTP 协议最初的设计思想得到了最大的发挥, 让人们真正理解 HTTP 本来面貌, 将对资源的获取、创建、修改和删除四种基本操作对应于 HTTP 协议提供的 GET、POST、PUT 和 DELETE 方法, 将方法统一, 降低了开发的复杂性和耦合度.

这种轻量级的 Web Services 架构风格, 它主要包括的概念有资源的可寻址性, 所有资源都设计唯一的 URI, 这种资源和 URI 的一一对应, 就使得 RESTful web Services 具有可寻址性. 还有无状态性、连通性及统一接口性等重要概念. 所以, RESTful Web Services 可以通过 Web 本身具有的松耦合特性以及其他分布式基础设施来实现具有很好交互功能和可伸缩性的 Web 服务. 因此, 无需引入复杂的 SOAP 协议, 比基于 SOAP 的 XML-RPC 更加简洁. 越来越多的 web 服务开始采用 REST 风格设计和实现. 例如, Amazon.com 提供接近 REST 风格的 Web 服务进行图书查找; 雅虎提供的 Web 服务也是 REST 风格^[4]. 因此, 应用该技术能够使得中医药行业资源易于描述、发现和整合.

2 信息共享与整合的 Web Services 解决方案涉及的主要内容

借助 Web Services 可以使不同系统数据信息共享, 解决信息孤岛问题同时也可整合不同系统数据得到二次数据.

2.1 中医药行业资源的注册与发布

为实现中医药行业资源的共享与整合, 首先需要中医药资源进行注册与发布. 为此创建资源服务登记管理服务器. 这样中医药行业资源根据需要可以在服务器上进行注册. 注册包括两个方面, 其中一个资源库基本信息包括: 库资源提供商, 资源 URI, 所属国家医药卫生分类标准编码, 库资源描述等, 另外一个资源库下面所包括具体的资源 URI、URI 帮助、URI 描述说明. 完成中医药行业库资源的登记注册后, 最重要的是对这些资源进行管理, 整合和发布, 让互联网用户真正受益于中医药行业资源. 可以直接发布, 即对服务提供商注册的各类资源信息直接发布, 按照

服务提供商设定的类别和填写的服务描述进行发布. 也可整合发布即是通过程序将现有服务提供商的资源进行重新整合, 创建新的资源, 完成新资源的定义和归属. 图 1 说明了资源注册与发布的主要过程. 用户通过 Internet 访问资源, 即可以通过资源服务登记管理服务器, 也可以单独访问每个中医药行业资源库, 无论“资源服务登记管理服务器”还是单独的中医药行业资源库均能够提供服务并且能够提供服务描述. “资源服务登记管理服务器”即注册中心, 提供给每个中医药行业资源进行服务登记注册, 记录每个资源库的基本描述信息, 服务提供点 URI; 同时对需要多个中医药资源库整合的数据信息也在此整合后进行新的资源命名, 提供新的服务; 另外应能够提供更加复杂的搜索查询访问. 用户(个人用户、系统用户)可到注册中心去查询资源, 也可直接访问各资源库.

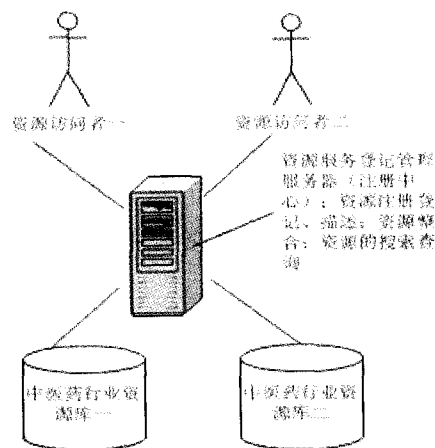


图 1 资源注册与发布

注册中心的实现是一个基于 Java Servlet 的系统. 已实现的功能有为各资源库提供资源注册、将各个资源库的资源信息汇总、解析用户查找的资源并匹配到具体的资源库、给具体的资源库发送请求获取对应的资源. 注册中心作为用户与具体资源库的中间管道.

2.2 中医药信息资源定义

面向资源的设计强调良好的 URI 定义应该具有描述特性, 使人们能直观的联系到该 URI 所代表的含义. 同时在设计 URI 时还应该使其具有一定的结构, 应赋予其一种统一的模式, 这样在客户端或者请求者对资源发起请求时, 可以很好的掌握对 URI 的使用^{[5][6]}. URI 的设计应该用路径变量(path variables)来表达层次

结构(hierarchy), 这样很容易通过添加路径变量来扩展层次结构。

在对中医药资源定义之前首先确定中医药行业的分类结构和分类编码。为此采用国家科学数据共享工程中中医药行业的分类和编码。中医药编码是 MM14。中医药数据分为: 中药信息、中医信息、针灸信息、古代文献和其他, 编码分别为: MM1411、MM1412、MM1413、MM1414 和 MM1499 共五个大类。这些资源的 URIs 设计如下, 用斜线 / 来分隔作用域信息, 以形成一个层次结构:

/中医药/中医药事业-----/MM14/MM1499
/中医药/中医信息 -----/MM14/MM1412
/中医药/中药信息 -----/MM14/MM1411
/中医药/针灸信息 -----/MM14/MM1413
/中医药/古代文献 -----/MM14/MM1414

这是一种很不错的 URI 设计, 它很容易通过添加路径变量来扩展层次结构, 例如“中药信息”分类下面的“中药材信息”资源 URI 可定义为: /MM14/MM1411/MM141111。若要进一步定义“中药材信息”下的“中药资源”, URI 可以是: /MM14/MM1411/MM141111/MM14111111。

所以, 当要提供下一级新的资源时, 只要把该层次架构往右延伸就行了。中医药资源的提供会有很多的单位。单位的域名作为该单位所提供资源 URI 的根。例如, 珠海博睿科技有限公司(<http://www.chinamtc.com>) 提供“中药材信息”资源, 那么该资源的 URI 可以是图 2 所示:

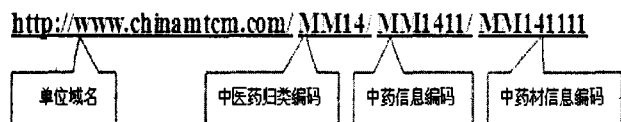


图 2 中医药资源 URI 设计

2.3 资源的表示

资源的表示可以是 html/xml/json 及其他媒体类型。xml 和 json 格式让整个 Web 逐渐地转化为一个机器可以理解、可以编程的平台, 人们能够将来自 Web 世界不同地方的资源聚合、转换变成满足各种新需求的数据或者服务。当然也可以以 html 格式提供给一般用户。

3 基于Restlet架构的Web Services 实现

下面给出一个中医药公司 Cmdata 公司(即一个中医药行业资源库)对其所提供资源的实现。该资源库一方面向注册中心注册了服务, 同时用户也可直接通过 URI 访问资源库的信息。技术实现采用 JavaEE。基于 Java 语言的 RESTful Web Services 架构有很多, 比如 Jersey、RESTeasy、Restlet 和 Spring mvc3。下面详悉说明如何应用 Restlet 架构实现 Web Services。

3.1 Restlet 架构简介

一个用 Restlet 框架构建的 RESTful 应用程序都需要两个基类: Application 和 Resource。逻辑上, Application 实例将 URI 映射到 Resource 实例。Resource 实例处理基本的 CRUD 命令, 当然, 这些命令都要映射到 GET、POST、PUT 和 DELETE。即是通过扩展 Restlet 框架的 Application 类来创建一个起点。在这个类中, 定义响应 URI 的 Resource。

由于基类 Application 是一个抽象类, 扩展类必须实现 createRoot()方法。在这个方法中, 可以创建一个 Router 实例, 并将 Resource 附加到 URI。Restlet 框架中的 Resource 类型被认为是 Restlet 类型。它们是使用 Restlet 框架开发的 RESTful 应用程序的核心。与 Application 类型不同, Resource 类并非抽象类。它更像是一个可以按需覆盖其默认行为的模版。

从较高的程度来看, Resource 有四个需要覆盖的方法。它们映射为基本的 HTTP 命令, 即 GET、POST、PUT 以及 DELETE。

3.2 构建应用系统

Restlet 框架极其轻便, 只用由几个核心 Restlet 基类扩展出来的少许类就能够构建一个功能完善的 RESTful 应用系统。图 3 是应用系统的分层结构图。

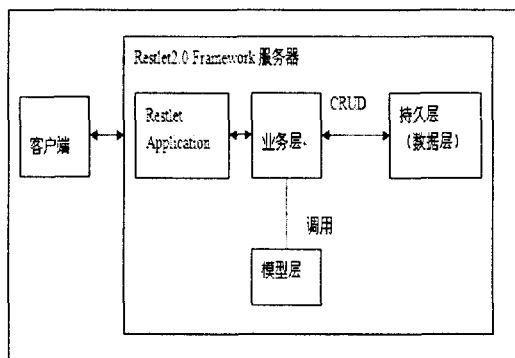


图 3 应用系统的分层结构

下面是对一个具体资源的请求实现,即请求某一药材信息资源,假定客户端请求“三七”这一药材信息,根据上面的介绍,请求药材的 URI 可以是: `http://localhost:8080/MM14/MM1411/MM141111/三七`,系统返回的表示是 JSON 格式,这个服务的实现过程如图 4 所示.客户端提出的资源请求被 Medicine Application 类接收到:

```
public class MedicineApplication extends Application {  
    @Override  
    public Restlet createRoot(){  
        Router router = new Router(getContext());  
        //MedicineResource 资源的 URI  
        router.attach("/{medicine}",MedicineResource.  
class);  
        return router;    }  
}
```

MedicineApplication 类继承于 Restlet 框架中的 Application 类,主要的作用是用来监听请求,并将请求分配给相应的资源去处理.

Router 类匹配资源类.系统已经从 Medicine Application 接收到了请求,接下来,它将根据一个 Router 类,把接收到的 URI 与资源进行映射,映射到 MedicineResource.class.

MedicineResource 调用资源类构造方法,创建一个资源类如下代码:

```
public class MedicineResource extends Resource{  
    //URI 地址中所包含的参数  
    String medicine = "";  
    String mTemp = "";  
    //资源 MedicineResource 的构造方法,主要是获取请求的方法类型和参数  
    public MedicineResource(Context context, Request request, Response response) throws UnsupportedOperationException {  
        super(context, request, response);  
        mTemp=(request.getAttributes().get("medicine")).  
toString();  
        medicine = URLDecoder.decode(mTemp, "UTF-8");  
        getVariants().add(new  
Variant(MediaType.APPLICATION_JSON));  
    }  
}
```

```
//GET 方法的处理函数  
@Override  
public Representation represent(Variant variant) {  
    //数据库相关操作,查询原始数据资源  
    String sql;  
    sql = "select * from medicine where name  
='"+medicine+"'";  
    System.out.print(sql);  
    Dao dao = new Dao();  
    ArrayList<HashMap<String, Object>> result = dao.  
getResult(sql, null);  
    dao.closeDB();  
    //将原始数据封装成 JSON 格式  
    JSONObject jo = new JSONObject();  
    jo.put("result", result);  
    //组装表示,包括内容和媒体类型  
    Representation representation = new JsonRepresenten  
tation(jo);  
    //返回表示  
    return representation;  
}
```

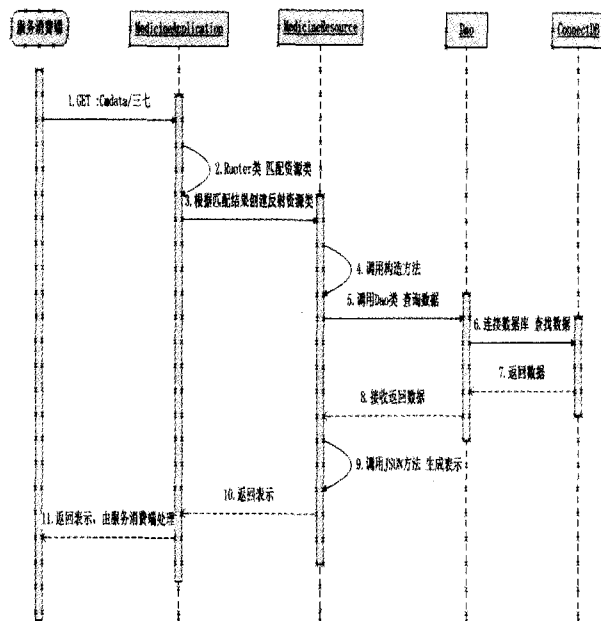


图 4 请求药材资源时序图

根据 Restlet 的 API 描述, Resource 类的作用主要是完成根据请求方法(GET、POST、DELETE、HEAD 等)对资源进行处理,最终返回一个表示给 Application

类。Representation 类的作用就是生成最终的表示, 返回给服务消费端。这种 JSON 数据格式有利于让 Web 逐渐地转化为一个机器可以理解, 可以编程的平台。即这个 Web Services 的 JSON 格式数据可供其它的计算机系统或程序使用, 例如搜索网站。对于 Cdata 也可以直接提供 HTML 格式的数据供一般用户查看。以上就是一次完整的资源请求。由此可以看到 Restlet 框架极其轻便, 只用由几个核心 Restlet 基类扩展出来的少许类就能够构建一个功能完善的 RESTful 应用程序。

4 结语

文中对 RESTful Web Services 的应用作了初步的研究与实现。还需进一步研究如何对数据资源进行整合。即除了能够实现资源的直接发布同时对于注册中心的服务器能够通过程序将现有服务提供商的资源进行重新整合, 创建新的资源, 完成新资源的定义和归属。同时也需要更深入的研究 REST 这种全新并且优秀的 web 设计风格如何为行业信息系统服务, 这是个复杂的问题。行业信息系统是个庞大复杂的系统, 有着不同的机构、不同类别的内容等。因此如何规划数

据集、如何将数据集转化为资源、如何命名资源、如何设计表示、如何把资源与资源间的联系表达出来即创建资源的连通性整合资源等很多问题需要研究。

参考文献

- 1 戴亚娥, 俞成海. 基于 REST 架构风格的 Web 2.0 实现. 计算机系统应用, 2009, 18(7): 165-168.
- 2 Fielding RT, Taylor RN. Principled Design of the Modern Web Architecture, ACM Transactions on Internet Technology (TOIT) (New York: Association for Computing Machinery), 2002, 2(2): 115-150.
- 3 Thomas FR. Architectural styles and the design of network-based software architectures [Ph.D Dissertation]. University of California, Irvine, 2000.
- 4 Leonard R, Sam R. RESTful Web Services, O'Reilly Media, Inc, 2007: 96-102.
- 5 McMillan R. A RESTful approach to web services. Network World, 2003(2): 20-24.
- 6 Sandoval J. RESTful Java Web Services. Packt Publishing. 2009: 78-99.

(上接第 232 页)

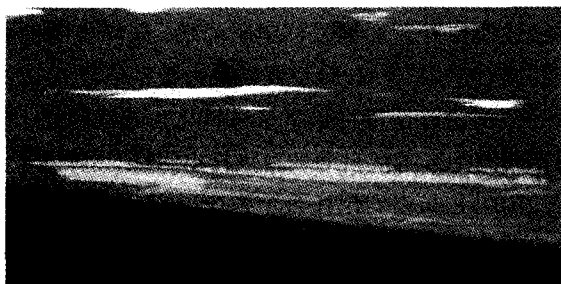


图 7 拍岸浪效果图

5 总结

本文基于 Vega Prime 仿真平台对海洋环境进行仿真, 将粒子系统引入到海洋环境中, 仿真出了比较逼真的海洋环境, 给人一种很强的沉浸感。可以很好的用于航海仿真, 虚拟海战场和游戏等。但是粒子系统的模型主要是依赖 Vega Prime 的特效模块, 相对比较简单, 没有考虑复杂环境的影响。自然现象和不规则物体的模拟一直是一项复杂的课题, 还需要更加深入

的研究。

参考文献

- 1 潘志庚. 虚拟现实及应用. 国际学术动态, 2009, 6: 22-24.
- 2 王乘, 李利军, 周均清, 陈大炜. Vega 实时三维视景仿真技术. 武汉: 华中科技大学出版社, 2005.
- 3 廖炎平, 刘莉等, 杜小菁, 戚泽华, 白海涛. Vega Prime 实时视景仿真中粒子系统的应用研究. 系统仿真学报, 2010, 22(4): 938-941.
- 4 孙鑫, 余安萍. VC++ 深入详解. 北京: 电子工业出版社, 2006.
- 5 王锐, 谷永山, 韦穗, 程鸿. 虚拟海洋场景的实时模拟研究与实现. 计算机工程与设计, 2012, 33(2): 695-699.
- 6 李苏军, 蒋杰, 杨冰, 吴玲达. 基于球面的海浪建模与绘制技术研究. 中国图象图形学报, 2009, 14(4): 744-752.
- 7 赵欣, 李凤霞, 战守义. 基于粒子系统的舰船航迹仿真. 计算机工程, 2008, 34(15): 22-24.