

# Jersey 的研究和在 Web 服务中的应用

陈一鸣<sup>1</sup>, 陈立南<sup>1,2</sup>

(1. 北京邮电大学 网络技术研究院, 北京 100876; 2. 北京信息科技大学 信息管理学院, 北京 100192)

**摘要:** 为了解决当前 Web 组件出现的可伸缩性差、接口通用性弱、交互延迟时间长的问题, 提出了轻量级的 REST 架构, Jersey 是 REST 架构中 JAX-RS 接口标准的实现, 通过研究它的连通性、可寻址性、无状态性以及稳定性和易用性等特点, 设计一套统一、高效、快速、方便访问服务器的 Client API, 实现为不同的智能移动终端提供统一的 Web 应用服务。

**关键词:** Jersey; JAX-RS; REST; Web 服务; 智能移动终端

**中图分类号:** TP393

**文献标识码:** A

**文章编号:** 1000-436X(2014)Z1-0156-04

## Research on Jersey and application in Web services

CHEN Yi-ming<sup>1</sup>, CHEN Li-nan<sup>1,2</sup>

(1. Institute of Network Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China;

2. School of Information Management, Beijing Information Science & Technology University, Beijing 100192, China)

**Abstract:** In order to solve the current Web components appear weak poor scalability, the generality of the interface, interactive delay time problem, put forward the lightweight REST architecture, the Jersey is the REST architecture - the realization of the JAX-RS interface standards, through the research of its connectivity, addressable, statelessness, and stability and ease of use features, design a set of unified, efficient, fast and convenient access to the server Client API, implemented as different intelligent mobile terminal with unified Web application services.

**Key words:** Jersey; JAX-RS; REST; Web services; intelligent mobile terminal

### 1 引言

由于互联网技术快速的发展, 现在主要有 2 种风格的 Web Services。一种是 XML-RPC(XML remote procedure call)即 XML 远程方法调用, 它以 SOAP<sup>[1,2]</sup>(simple object access protocol)协议为信息传递的基础, 采用复杂的嵌套协议不仅给开发设计带来不必要的麻烦, 也使得不同系统间的交互产生阻碍。另一种是 REST 即表述性状态转移, 使用 HTTP 协议语义和语法, 运用一组可寻址的 URI 资源向客户提供 Web Service。由于 Web 本身具有的松耦合特性, 再加上其分布式的基础设施可以提供具有可伸缩性和很好交互性的 Web 服务。

JAX-RS<sup>[3]</sup>(Java API for REST Web services)是

构建 REST 服务中一个固定的接口, Jersey 是 Sun 公司对 JAX-RS 接口的参考实现, 它在稳定性、易用性和 Spring、Hibernate 等其他框架的支持度方面超过其他几种实现, 成为构建 Web 服务的首选。而且现在智能移动终端发展非常迅速, 在提供丰富多彩应用的同时也成为人们日常生活中不可缺少的移动设备, 所以本文在研究 Jersey 的同时, 利用 Jersey 开发基于移动终端统一的应用服务, 不仅有一定的学术意义, 也非常符合社会的需求。

### 2 REST

REST<sup>[4]</sup>(representational state transfer)是在 Roy Thomas Fielding 博士论文首次被提出。REST 是将事物抽象为资源, 而且把唯一的资源标识符 URI 赋

收稿日期: 2014-10-18

基金项目: 北京市教委科研计划基金资助项目(KM201311232017)

Foundation Item: Beijing Municipal Commission of Education Research Project (KM201311232017)

予一个资源,然后通过 HTTP 处理和传输资源状态。REST 是一组架构约束条件和原则,只要满足这些原则和约束条件的设计或应用程序就是 RESTful。RESTful Web 服务是符合 REST 风格的轻量级 Web 服务,其主要特点如下: 1) 所有资源都有唯一 URI 作为识别符; 2) 使用标准方法; 3) 资源多重表述; 4) 无状态通信。

### 3 JAX-RS 与 Jersey

JAX-RS<sup>[5]</sup>(Java API for RESTful Web services)旨在定义一个统一的规范,可以运用一系列固定的接口开发 REST 应用,避免了依赖于第三方框架。同时, JAX-RS 使用 POJO (plain old java object) 简单的 Java 对象编程模型和基于标注的配置,可以有效缩短 REST 应用的开发周期。目前 4 种 REST 架构 JAX-RS 接口标准的实现分别如下<sup>[6]</sup>。1) Restlet。它抛弃了 Servlet API,自己实现了一套 API,能够支持复杂的 REST 架构设计,但是性能相对很差,而且集成难度高。2) CXF。它是 XFire 和 Celtix 的合并,支持 Web Services 标准,通用性强,但是文档不丰富,学习起来较困难; 3) RESTEasy。提供多种框架构建 RESTful Web Services 和 RESTful Java 应用程序,但是在安全方面存在明显的不足。4) Jersey<sup>[7]</sup>。它是 Sun 公司发布的 JAX-RS 接口的参考实现, Jersey 实现了 JAX-RS 中对标注的支持,使得使用 Java 开发 REST 架构的 Web 服务更加容易。在 Sun 的支持下, Jersey 从集成能力、稳定性、性能、产品成熟度、安全性、易用性综合起来较另外 3 种方法更强,应用也越来越广泛。

Jersey 是 SUN 工作提供的对 JAX-RS 的实现,如图 1 所示,主要包括以下 3 个部分。

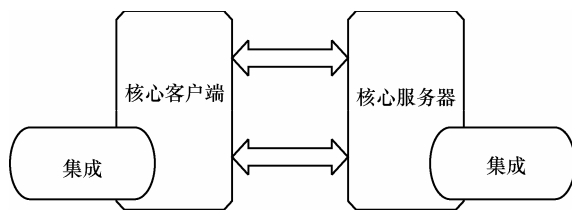


图 1 Jersey 核心组件结构

1) 核心服务器(core server): 通过提供 JAX-RS 中标准化的标注(annotation)和 API, 以便捷的方式开发 RESTful Web 服务。

2) 核心客户端(core client): Jersey 客户端 API 帮助用户与服务端建立通信连接。

3) 集成(integration): Jersey 可以提供轻松集成 Spring、Guice、Apache Abdera 的类库,能够添加 REST 客户端,支持 GlassFish、NetBeans6.5、Maven2 等产品级别的系统集成,并可以通过 jersey-json.jar 实现对 JSON 的支持。

Jersey 提供的标注将资源类及其对应的方法封装为 Web 资源,主要注解包括: @Path: 标注资源类或方法的相对路径; @POST、@GET、@PUT、@DELETE: 标注方法是用 HTTP 请求的类型,当客户端向代表某个 Web 资源的 URL 直接发送 HTTP 请求时, JAX-RS 运行时会被该类型注解的方法来处理请求; @Consumes: 标注可接受的 MIME 媒体类型; @Produce: 标注返回的 MIME 媒体类型; @PathParam、@QueryParam、@HeaderParam、@CookieParam、@MaxtrixParam, @FormParam 分别标注方法的参数来自于 HTTP 请求的不同位置,例如 @PathParam 来自 URL 的路径, @QueryParam 来自于 URL 的查询参数, @HeaderParam 来自 HTTP 请求的头信息, @CookieParam 来自于 HTTP 请求的 Cookie。

### 4 Jersey Web 应用的设计与实现

#### 4.1 整体分析

本系统设计一个航班管理应用来展示 Jersey 在智能移动终端的应用。如图 2 所示,该应用使用 C/S 模式, Web 服务器使用分层设计方法,利用 Jersey 技术的 Service 层为智能移动终端提供统一的访问接口,智能移动终端利用 Http 协议访问给定 URL 与服务器进行数据交互。

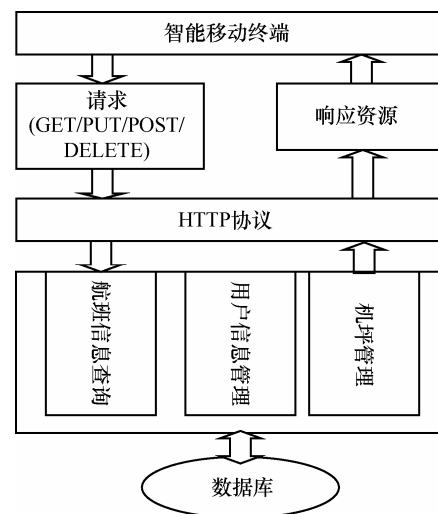


图 2 Jersey 核心组件结构

### 4.2 资源的设计及映射

由于在 Jersey 中，所有的资源都有一个 URI，一般都是使用逻辑的 URI 进行设计，这样避免了因为服务器物理 URI 资源的修改影响客户端正常的访问。URI 设计遵循 RFC-2616 所定义的协议方式，建立创建、检索、更新和删除(CRUD, create, retrieve, update and delete) 操作与 HTTP 方法之间的一对一映射，即：创建资源(POST)、检索资源(GET)、更改资源(PUT)、删除资源(DELETE)。本应用的 URI 映射关系如表 1 所示。

表 1 资源与 URI 映射关系

模块	Path = http://serverIp:Port/rest/		
	子功能	Id	操作
航班信息查询	进港航班	{Flight/flightIn}	POST
	出港航班	{Flight/flightOut}	GET PUT
用户信息管理	权限分配管理	{User/authority}	
	机场工作人员信息管理	{User/worker}	DELETE
	旅客信息管理	{User/passenger}	
停机坪管理	舱门管理	{Apron/gate}	
	站台管理	{Apron/station}	POST
	机位管理	{Apron/slot}	GET PUT
	航空油车管理	{Apron/tanker}	

### 4.3 服务器端设计

服务器端运用 Jersey+Spring+Hibernate 技术实现 MVC 的分层开发方案。Jersey 和 Struts 很相似，但是它比 Struts 更轻量级，更方便。Jersey 技术作为控制层来实现服务器和客户端的数据交互，通过接受客户端数据请求来操纵 Hibernate 获得数据，并把数据封装送给客户端。主要代码如下。

```

@Context
UriInfo uriInfo;
@Context
Request request;
@Context
HttpHeaders headers;
@GET|POST|PUT|DELETE //表示只接受请求的形式
@Path("/{id}") //接受请求的路径 id
@Produces(MediaType.APPLICATION_JSON)/
返回数据类型
    
```

```

public String getFlight(){
    ApplicationContext ctx = new
    ClassPathXmlApplicationContext("beans.xml");
    FlightService = (FlightService) ctx.getBean
    ("FlightService");//创建 service 层对象
    .....//业务逻辑处理
    return json = objectMapper.writeValueAsString
    (list)//把 List 转换为 JSON 字符串送给客户端
}
    
```

### 4.4 移动客户端设计

由于客户端和服务器是通过 HTTP 协议进行数据交互的，所以只要智能移动终端具有 HTTP API 就可以进行交互，如 Android (HttpClient)、微信(Curl)、Python(httplib)，可以满足现在市面上所有的移动客户端，而且 Jersey 设计一套 Client API 可以高效、快速、方便的访问服务器，程序如下。

```

Client c = Client.create();
WebResource r= c.resource("http://ServerIP:Port
/rest");//创建资源
ClientResponse response;//http 返回的数据对象
response=r.path(id).type(MediaType).get (Cli
entResponse.class);//发起 GET 请求并接受回传数据
对象
response=r.path(id).type (MediaType).post (Cli
entResponse.class,form); //发起 POST 请求并接受回传
数据对象
response=r.path(id).accept(MediaType).put
(ClientResponse.class, c); //发起 PUT 请求并接受回
传数据对象
response=r.path(id).accept(MediaType).delete
(ClientResponse.class, c); //发起 Delete 请求并接受
回传数据对象
Object entity = response.getEntity(Object.class);
//得到服务器返回的数据对象
    
```

## 5 结束语

本文通过对基于 REST 架构 Jersey 的阐述和分析，利用 Jersey 为智能移动终端开发 Web 服务，并成功实现数据访问，而且该应用已经在国内的某个机场成功部署。实践证明，Jersey 有很好的方便性和稳定性，它在移动服务领域有很大的发展潜力。

## 参考文献:

- [1] RESTfulWebservice 和 SOAP Webserivce 对比及区别[EB/OL]. <http://www.blogjava.net/diggbag/articles/361703.html>.  
Diffence between RESTful Web service and Web service[EB/OL]. <http://www.blogjava.net/diggbag/articles/361703.html>
- [2] THOMAS R. Fielding Architectural Styles and the Design of Network-based Software Architectures[M]. Doctoral DisSertation, Dept of Computer Science, University of Californis, Irvine, 2000.
- [3] 谢玉开. 基于 JAX-RS 的面向资源架构应用研究[D]. 浙江理工大学, 2011.  
XIE Y K. Application of resource-oriented architecture based on JAX-RS[D]. Zhejiang Sci-tec University, 2011.
- [4] 黄羽中. REST 开发框架纵览[J]. 软件世界, 2007,11:40-41.  
HUANG Y Z. REST Development framework overview[J]. Software in the World, 2007, 11:40-41.
- [5] JAX-RS: The Java™ API for RESTful Web service[EB/OL]. <http://jcp.org/en/jsr/detail?id=311>.
- [6] 丁晶晶, 饶庆云, 苏乐乐等. 基于 Jersey 的空间分析 REST API 设计与实现[J]. 测绘与空间地理信息, 2011, 34:21-26.

DING J J, RAO Q Y, SUN L L, *et al.* Jersey-based spatial analysis REST API design and implementation[J]. Surveying and spatial information, 2011, 34:21-26.

- [7] Oracle. Jersey[EB/OL]. <https://jersey.java.net/>.

## 作者简介:



陈一鸣(1990-), 男, 河南平顶山人, 北京邮电大学硕士生, 主要研究方向为面向对象和组件的软件。



陈立南(1972-), 女, 黑龙江勃利人, 北京邮电大学博士生, 北京信息科技大学副教授, 主要研究方向为新一代互联网关键技术等。