

解析 Restful Web Service 架构

赵博文

(同济大学软件学院, 上海 201413)

摘要:本文展示了 Http 协议的强大能力, 如何定义什么是 Restful Web Service 架构以及以当今 RPC 式 Web 服务的对比, 并解析了 Restful Web Service 架构的四个特征: 可寻址性、无状态性、连通性和统一接口。

关键词: HTTP; URL; REST; Web 服务

中图分类号: TP393.04

文献标识码: A

Analysis of Restful Web Service Architecture

ZHAO Bo-wen

(School of Software Engineering, Tongji University, Shanghai 201413)

key words: HTTP; URL; REST; Web Service

1 Http 协议基础

Http 是一种基于文档的协议。客户端把文档放在 Http 信封里, 然后发给服务器; 服务器作为回应, 把响应文档放到 Http 信封里, 然后发回给客户端。

以下是一个 Http 请求:

GET /index.html HTTP/1.1

Host: www.google.com

Accept: text/html

Accept-Charset: utf-8

Connection: keep-alive

Http 请求信息包括以下几个部分:

HTTP 方法: 对于这个请求是 "GET" 方法。Http 方法的名称, 就如同编程语言的方法名称, 表示客户端希望服务器如何处理这封信。

路径: 是 URI 主机名后面的信息。对于这个请求, URI 是, 路径是 www.google.com。路径就是信封的地址。

请求报头: 是一组关键字-值对, 起元数据的作用, 例如向主机、接受字体等元数据信息。

实体主体或表示: 就是信封里的文档。在这里 GET 信封里没有文档, 是空的。

Http 响应信息包括三个部分。Http 响应代码, 通知客户端的请求是成功还是失败; 响应报头, 跟请求报头一样, 起元数据的作用; 实体主体或表示, 相当于信封里具体的信件文档。

2 Web 服务架构的分类

Http 是 Web 服务所共有的, 而 Web 服务之间的差别, 是因为它们在两个问题上的做法不同:

一是客户端是如何把自己的意图传达给服务器的, 具体地说就是客户端如何告诉服务器应该做什么样的操作。例如是获取数据、删除数据还是改写数据呢? 这种对数据采取什么操作的信息, 称之为方法信息。

二是客户端如何告诉服务器对哪些数据进行操作, 例如服务器已知客户端请求删除数据, 但要删除那些数据呢? 这种信息称为作用域信息。

2.1 传统的 RPC Web 服务架构

RPC (远程过程调用) 式 Web 服务通常从客户端收到一个充满数据的信封, 然后返回一个充满数据的信封。常见的信封格式是 SOAP。RPC 架构意味着方法信息和作用域信息都放在信封实体里。

例如一个描述 XML-RPC 请求的 HTTP 信封:

POST /RPC HTTP/1.1

HOST www.search.com ...

<?xml version="1.0"?>

<method>

<methodName>search</methodName>

</method>

在上述的 RPC Web 服务调用中, 无论你向服务器请求什么方法, URI 永远是 www.search.com, Http 方法永远是 POST。

2.2 Restful Web 服务架构

Restful Web 服务是符合 REST (Representational State Transfer) 风格的 Web 服务架构, 按照 Roy Fielding 博士论文里的评判标准是符合 Web 特点的。

Restful Web 服务架构意味着, 方法信息都在 HTTP 方法里; 作用域信息都在 URI 里。

方法信息放到 HTTP 方法里就是指通过 GET、PUT、DELETE 等标准的 HTTP 方法来区分获得数据、修改数据或删除数据。使 web 服务有统一的接口。不像 RPC 式的服务将方法信息放到实体里, 无论客户端请求获得数据或删除数据都使用 POST 方法, 不具有统一接口, 使 web 服务之间的交互困难。

作用域信息放在 URI 里就是指无论要操作什么样的数据都通过 URI 获得, 这样一个 Restful Web Service 的请求信息的第一行就能了解客户端要做什么了。

例如: GET /Report/week HTTP/1.1 就是要获得周报表数据。而 RPC 式则将作用域数据放到实体里, 使得 RPC 式的客户端请求格式复杂同时也不具有可寻址性 (后面会详细介绍)。

3 Restful Web 服务架构的四个特征

这里我们将阐述 Restful Web 服务架构的四个特征: 可寻址性、无状态性、连通性和统一接口。

3.1 Restful Web 服务架构的资源概念

任何事物, 只要具有被引用的必要, 它就是一个资源。在 Web 服务上它至少有一个 URI (统一资源标识符), URI 既是资源的名称, 也是资源的地址。在 Restful Web 服务架构中 URI 只能指示唯一的一个资源, 但是响应信息可以包括多个资源, 例如包括一些其它资源的连接信息。

3.2 可寻址性

可寻址性是指用户可以通过唯一标示的信息获得特定的资源。例如在 windows 浏览器中, 你要察看哪个文件夹直接敲入地址, 即可直接访问。例如: C:\Program Files\Windows NT 就可直接访问 Windows NT 文件夹。在 Web 服务中可寻址性就是指每一则服务都是可直接访问的, 可以被添加到收藏夹里。(下转第 7 页)

利用它来学习的要求,这也有助于高职学生掌握操作计算机及网络的技能提高。使学生真正成为社会需要的技能型人才。

3 现代教育技术给高职教育带来的负面影响

现代教育技术应用带来了教学效果和教学质量的提高,其中许多突出特点在实践中已显示了强大优势。但是任何事物的发展总是一个辩证的过程,在现代教育媒体开发与应用中也出现一些不容忽视的问题,给高职教育教学带来了一些负面影响。

- (1)多数多媒体教学课件重外表轻实用;
- (2)过分依赖多媒体设备和技术;
- (3)互联网成为毒害学生的工具;
- (4)导致新的文化入侵。

4 消除现代教育技术给高职教育带来负面影响的策略

现代教育技术带给高职教育的种种负面影响主要是人们对它的误用或滥用造成的,因此随着全民素质(包括教师素质)的提高,逐步消除“误用”或“滥用”的人为基础,一些负面影响将随之消失。

注意把学生的网上活动吸引到健康有益的内容上,继承我们民族的美德,汲取世界文化的精华。

在学生中加强网上道德教育,倡导诚实、信任、尊重他人权利、维护公共信息安全、拒绝和抵制不正当或负面信息等健康的道德观念和行,注意辨别信息的真与假、美与丑、善与恶,从我做起,努力营造良好的网络环境。

随着科学技术的发展,人类利用现代教育技术的巨大作用逐步实现高科技水平与高人文素质的统一,从而消除因教育技术本身的不足所带来的一些负面影响,并且消除负面影响的能力必将随技术本身的进步而有所增强。

总之,应用现代教育技术,促进了高职教育的发展和改革,当然,我们也必须注意到研究现代教育技术可能会给高职教育带来的负面影响,因此,我们要防微杜渐,提出切实有效的防范策略,使高等职业教育健康顺利地发展。

参考文献:

- [1]章伟民.教育理念与教育技术[J].《电化教育研究》,2002(4):3-7.
- [2]卓发友.正确理解现代教育技术的涵义[J].《电化教育研究》,2002(5):9-11.
- [3]吴疆.王润兰编著《21世纪现代教育技术》[M].北京:人民邮电出版社,2001(11).
- [4]刘儒德.信息技术与教育相整合的进程[J].高等师范教育研究,1999(9).
- [5]祝智庭.教育信息化:教育技术的新高地[J].中国电化教育,2001(2).
- [6]徐晓东.存在万能的教育技术学基础理论吗[J].电化教育研究,2004(9).
- [7]应云飞.职业教育教师存在的问题及改进措施[J].成功(教育),2007(7).

(上接第4页)Web是可寻址的似乎是理所当然的,可是许多Web应用却不是可寻址的。例如Gmail在线Email服务,Gmail的URI始终是http://mail.google.com/。

无论用户是什么操作,是要获得信息还是要上传信息,用户都不会看到其它URI,例如“关于狮子的邮件”这个资源就不是可寻址的。

在Restful Web服务中如果一个应用把资源发布出来,那么该应用就是可寻址的。因为资源是通过URI暴露的,所以一个可寻址的应用应该会为它提供的每一则信息都发布一个URI,URI的数量在Restful web Service中是无限的。可寻址性的优点是可以让客户端灵活自如使用Web Service上资源,甚至超出Web服务设计者的想象。可寻址性本身就是Web上的一大优点,在Restful web Service中我们应当保留这个优点。

3.3 无状态性

无状态性意味着每个Http请求都是完全孤立的。当客户端发一个Http请求,请求里应当包含服务器实现该请求的所有信息,不依赖于任何以前的信息。假设本次请求需要之前请求的信息,那就应当将该信息也放到本次请求里。

Restful Web Service应当遵循这样的原则,从可寻址性方面考虑无状态性。无状态性要求:服务器可能的状态也是资源,也应该有自己的URI。客户端不必为了某个资源的请求而让服务器提前进入某种状态。例如google搜索网站,打开http://www.google.com/后在输入框填入“apple”,然后点击Search,得到相应结果。用户同样可以直接输入http://www.google.com/search?hl=en&q=apple&aq=f得到搜索结果而不用先打开google主页,这就是无状态性的一种体现。

3.4 连通性

表示(Representation)就是序列化的数据结构,把其中的数据取出即可丢弃。但在Restful Web Service中,表示应该是超媒

体(HyperMedia),也就是说表示中不仅包含数据,还包含指向其它资源的连接。这种具有“连接”的特性,称为连通性。

连通性使服务器通过超媒体告诉客户端当前状态有哪些后续状态可以进入。例如在http://www.google.com/search?hl=en&q=apple&aq=f资源里的“下一页”连接起的就是这种推进作用——它指引你如何从当前状态进入下一个可能的状态。在Restful Web服务的文档中只要包含URI就可以指向本应用的其他状态,因为Restful Web Service是可寻址且无状态的应用。

3.5 统一接口

在Restful Web Service中,我们对资源的操作仅限制为HTTP提供的基本操作,这里列举常用的四种方法:

“获取资源的一个表示:Http GET。

“创建一个新资源:向一个新的URI发送HTTP PUT或向一个已有URI发送HTTP POST。

“修改已有资源:向已有的URI发送HTTP PUT。

“删除已有资源:HTTP DELETE。

REST要求使用统一的接口,即每个服务器都以同样的方式使用HTTP接口。这样要求的意义在于:不论你对Web上哪个资源作GET请求,都意味着读取。统一接口令服务之间的交互更加容易。

参考文献:

- [1]Roy Thomas Fielding Architectural Styles and the Design of Network-based Software Architectures Computer Science 2000.
- [2]Fielding R T. Principled Design of the Modern Web Architecture. Proc. of the 2000 Intl.Conf.on Software Engineering (ICSE 2000), Limerick, Ireland. 2000- 06:407- 416.
- [3]Leonard Richardson Restful Web Service[M].电子工业出版社 ISBN 978-7-121-06227 2008-5.