

Application of RESTful Web Service in Mobile Device

Shoujian Yu, Cong Wei, Peng Huang

School of Computer Science and Technology, Donghua University, Shanghai
Email: jackyyshj@dhu.edu.cn, liujianggufan@163.com, 392817498@qq.com

Received: Jun. 14th, 2014; revised: Jul. 22nd, 2014; accepted: Aug. 1st, 2014

Copyright © 2014 by authors and Hans Publishers Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY).
<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

In order to solve the problem that the traditional PC-based system architecture is not compatible with the mobile system when we implement a mobile application of the corresponding PC version and the problem of speed, performance, efficiency and usability of it, we proposed a RESTful web service to deal with the issue of heterogeneous platform interoperability and information sharing, which is more compact and more usable compared with the traditional SOAP and RPC frames. Because of the limited resources of the mobile device, we use JSON as data exchange format. In the end, we implement a mobile terminal application of a document management system in both iOS and Android terminal in combination of a company's actual demand, which uses the RESTful web service framework.

Keywords

REST Web Service, JSON Format, iOS System, Android System, Document Management System

REST风格的Web服务在移动终端的应用

于守健, 魏 聪, 黄 鹏

东华大学计算机科学与技术学院, 上海
Email: jackyyshj@dhu.edu.cn, liujianggufan@163.com, 392817498@qq.com

收稿日期: 2014年6月14日; 修回日期: 2014年7月22日; 录用日期: 2014年8月1日

摘要

针对传统的基于PC端的系统在开发其对应的移动版本的时候遇到的系统架构不兼容以及速度、性能、效率和易用性的问题，本文提出了基于REST风格的Web服务来解决异构平台互操作和信息共享的问题，该框架比传统的SOAP和RPC框架更加简洁易用。由于移动终端资源有限，故本文采用JSON作为数据交换的格式。最后，本文结合某公司的一个文档管理系统的需求在iOS终端和Android终端实现了一个采用REST风格的Web服务框架的移动终端的应用。

关键词

REST Web服务, JSON格式, iOS系统, Android系统, 文档管理系统

1. 引言

随着移动互联网的发展以及智能手机和平板电脑的普及，通过移动终端上网的人数越来越多，移动互联网的规模正在急速增长。据中国互联网络信息中心 2012 年 7 月《中国互联网络发展状况统计报告》显示：截至 2012 年 6 月底，我国手机网民规模达到 3.88 亿，较 2011 年底增加了约 3270 万人，网民中用手机接入互联网的用户占比由上年底的 69.3% 提升至 72.2%。因此，对于传统的基于 PC 的应用系统，开发其对应的移动版本的需求也与日俱增。但是，将现有系统移植到移动终端存在很多问题：移动终端平台与传统的基于 PC 的系统在系统架构、硬件平台、开发工具等方面存在异构，而且移动终端的资源有限，比如电池、带宽、流量、存储容量等。

目前，国内外企业针对异构平台互操作和信息共享问题主要是通过 Web 服务来解决。但是，当前包括 SOA 在内的大量 Web 服务架构均采用 RPC 风格构建，在 Web 级的大规模应用中 RPC 风格的架构在扩展性、性能等方面存在着瓶颈[1]。

本文提出了一个基于 REST 架构风格的 Web 服务系统框架，并且结合某公司基于 PC 的文档管理系统在当前主流的两种移动终端操作系统——iOS 和 Android 平台上实现了其对应的简化版本，使得异构的平台可以共享数据资源，而且采用了轻量级的数据交换格式 JSON 在不同平台间交换数据，有效地减小了带宽和流量，并且描述了系统的架构和实现的关键技术。

2. 相关工作

2.1. REST 概述

REST(Representational State Transfer)最早由 Roy Fielding 在他的博士论文[2]中提出，描述了 Web 作为一个分布式超媒体的应用，相互链接的资源通过交换代表资源状态的表述来进行通信[3]。要理解 REST，就必须澄清资源(Resource)、表示(Representation)和状态(State)三个概念[1]。资源可以是任何事物，它可以是一个实物，也可以是一个抽象的概念。只要有被引用的必要，就可以将其抽象为一个资源。表示是一个资源当前状态的有用信息，对于给定的资源，可以有多个不同的表示。REST 中的状态分为资源状态和应用状态：资源状态是关于资源的信息，保存在服务端；应用状态是客户端在应用中所处状态的信息，由各个客户端自己维护。

REST 系统中的组件必须遵守下列约束[2]：

- 1) 所有的资源都是通过 URI 来标识；

- 2) 组件的相互操作是无状态的;
- 3) 资源的操作要通过对它的描述进行;
- 4) 自描述的消息;
- 5) 超媒体是程序状态的驱动器;
- 6) REST 强调中间媒介的作用(包括代理服务器、缓存服务器和网关)。

在 REST 系统中,所有资源都有一个 URI,包括 Web 服务也可以用 URI 来标识。HTTP 提供了 GET、POST、PUT 和 DELETE 四种基本的方法用于获取、创建、修改、删除资源四种最常见的操作[4]。在此,HTTP 作为一个程序协议来使用,而不是只作为传送一个 SOAP 消息的传输协议来使用[5]。

2.2. JSON 概述

JSON(Java Script Object Notation)是一种轻量级的数据交换格式[6]。易于人阅读和编写,同时也易于机器解析和生成。JSON 采用完全独立于语言的文本格式,但是也使用了类似于 C 语言家族的习惯,这些特性使 JSON 成为理想的数据交换语言。

JSON 建构于两种结构[6]:

1) “名称/值”对的集合。不同的语言中,它被理解为对象(object),纪录(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)。例如: {"firstName": "Brett", "lastName": "McLaughlin"}。

2) 值的有序列表。在大部分语言中,它被理解为数组(array)。例如,假设希望表示一个人名列表,就只需将多个带花括号的记录分组在一起。例如:

```
{ "people": [ { "firstName": "Brett", "lastName": "McLaughlin", "email": "brett@sina.cn" },
  { "firstName": "Jason", "lastName": "Hunter", "email": "jason@163.com" },
  { "firstName": "Elliotte", "lastName": "Harold", "email": "elliotte@sohu.com" }
]}
```

在这个示例中,只有一个名为 people 的变量,值是包含三个条目的数组,每个条目是一个人的记录,其中包含名字、姓氏和电子邮件地址。

3. REST 风格的 Web 服务系统框架

REST 风格的 Web 服务系统是一个 C/S 架构的系统,客户端和服务端通过 HTTP 协议传递数据。客户端和服务端的交互是无状态的。服务端主要包括以下几个部分: REST API 模块,数据封装模块,业务逻辑模块,数据管理模块和日志管理模块。具体结构如图 1 所示。

3.1. 主要模块介绍

REST API 模块用来描述系统资源。在 REST 系统中,所有资源都是用一个 URI 标示。REST API 是客户端与服务端交互的接口。所有的业务逻辑在客户端封装好后都是通过 REST API 被客户端调用。

数据封装模块用来将数据封装成目标格式。一般 Web 服务会将数据封装成 XML 或者 JSON 格式。由于 XML 格式的比较复杂,解析的代价比较高,而且本文针对的客户端是移动终端,移动终端的带宽、流量等资源有限,故本文将数据封装为 JSON 格式。

业务逻辑模块主要功能就是实现具体的业务逻辑。

3.2. 系统工作机制

系统工作机制如图 2 所示。用户发出一个请求后,移动终端通过相应的 REST API 提交请求,然后

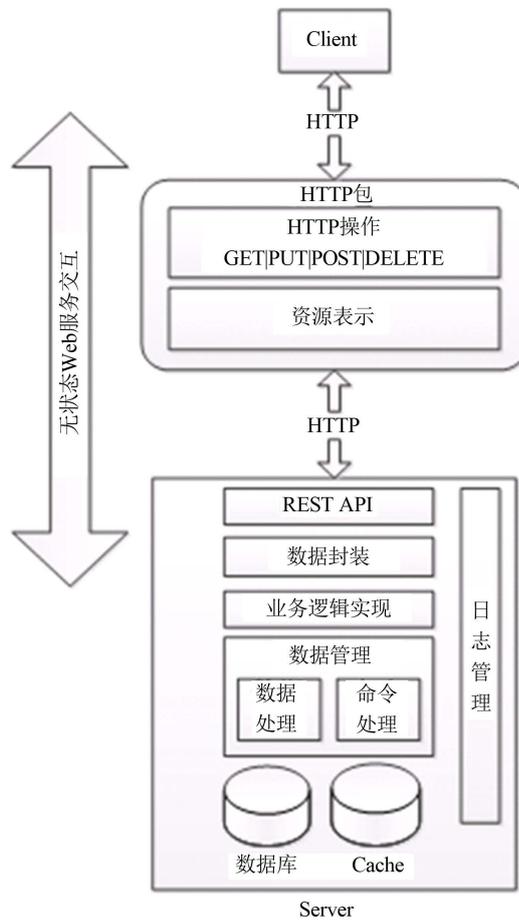


Figure 1. System framework
图 1. 系统框架

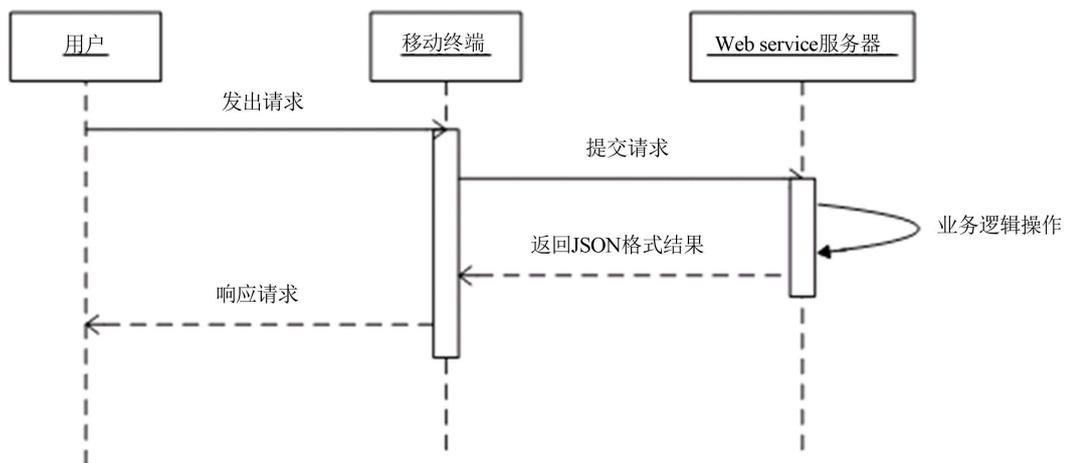


Figure 2. System sequence diagram
图 2. 系统顺序图

Web Service 服务器根据提交的请求执行相应的业务逻辑操作，接着将请求的结果封装成 JSON 格式返回给移动终端，最后移动终端会响应用户的需求。

4. 原型系统实现

4.1. 系统架构

REST 风格的 Web 服务系统由数据库服务器、Web Service 服务器、iOS 终端和 Android 终端组成。该系统采用分层的体系结构，数据层用来存储数据，业务逻辑层用来处理实际的业务，并且给移动终端提供 REST 风格的 API，而展示层通过具体的 iOS 终端和 Android 终端向用户提供图形化的操作界面，展示系统功能。移动终端通过移动通信网络统一与 Web 服务交互，操纵数据库里的数据。系统架构具体如图 3 所示。

4.2. 系统实现

4.2.1. 系统功能

该文档管理系统移动终端实现的主要功能为提供移动环境下的信息浏览、登录、文档检索、信息收藏、实体服务进度监控等功能，是 PC 门户在移动设备上的一个简化版。因此，移动终端主要包括门户首页及登录模块、文档检索模块、我的收藏模块、我的调卷模块、我的订阅模块和更多六个部分组成。

4.2.2. 资源描述方法

在 REST 风格的 Web 服务系统中，任何需要被引用的事物都可以视为资源。而对于文档管理系统，资源就是数据库中的数据，而这些资源是通过 URI 的形式定义资源标识。对于本系统，每一个对数据库的操作都会映射成为一个 URI，移动终端通过这些 URI 访问系统的资源。

例如，对于登录功能来说，其对应的 URI 为 `http://192.168.1.1:8080/DMS/user/login.json`，移动终端通过这个接口访问数据库中的资源，并且传递给 Web 服务器 JSON 格式的参数，Web 服务器端通过访问数据库中的数据，判断是否为合法的用户，并且响应客户端的请求，返回给客户端 JSON 格式的结果，例如：`{"result": "1"}` 表示该用户为合法用户，可以登录系统。

4.2.3. REST 接口实现

在定义 REST 风格的 Web 服务时，除了需要定义访问资源的 URI，还需要指定传入参数和返回结果的数据格式以及访问方式。由于移动终端的资源有限，故我们采用轻量级的数据交换格式 JSON，指定传入和输出参数都为 `application/json`。

访问方式使用 HTTP 提供的四种基本方法[4]，即：

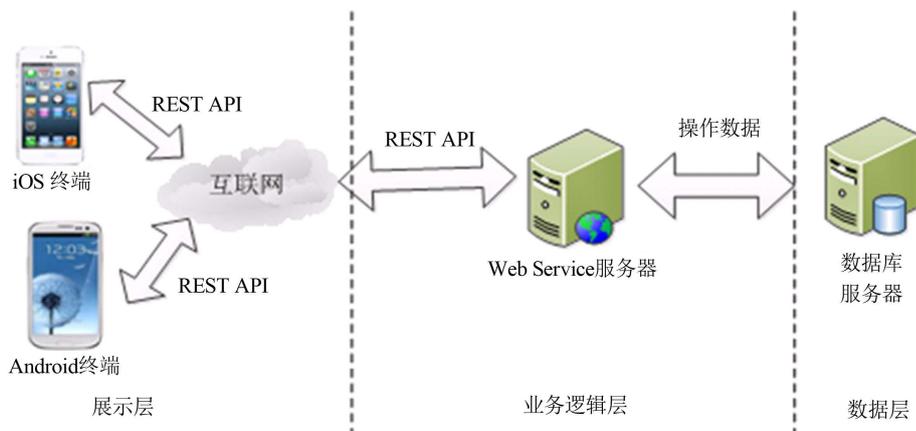


Figure 3. Document management system architecture
图 3. 文档管理系统架构

获取资源的一个表示：HTTP GET。如查看文档的描述信息。

创建一个新资源：向一个新 URI 发送 HTTP PUT，或者向一个已有 URI 发送 HTTP POST。如将一个文档添加到收藏夹。

修改已有资源：向已有 URI 发送 HTTP PUT。如修改用户的联系方式。

删除已有资源：HTTP DELETE。删除一个收藏的文档。

4.2.4. iOS 终端实现

iOS[7]终端实现 REST 风格的 Web 服务主要是通过 RestKit。RestKit 是一个基于 Objective-C 的 iOS 框架，其目的是为了使 RESTful Web 服务更简单、快速和有趣[8]。它基于强大的对象映射系统并且使用了一个干净、简单的 HTTP 请求/响应 API，减少完成任务所需的代码量。

RestKit 除了发送请求与接受响应这些基本功能外，还附带 Core Data，以及将远程 JSON 映射为本地对象的功能。首先，RestKit 要求我们在客户端定义一个 Model 类，用于存储数据和处理业务逻辑。远程 JSON 中各键将直接映射为 Model 中各成员变量。对应每一个模型都要定义一个对应的 Model 类，这是强制的。强制定义模型带来的好处是我们只需要写一次字典中的 key，不再会因为字典的 key 拼错导致程序崩溃了。

iOS 终端的界面如图 4 所示，用户可以通过页面大图标或是页面底端的标签进行文档管理系统相应的操作。

4.2.5. Android 终端实现

在 Android[9]终端则直接通过 org.apache.http 包下提供的 HttpResponse、HttpEntity 类以及 org.apache.http.client.methods 包下的 HttpGet 类等与 Web service 服务器交互。通过 JSONObject 和 JSONArray 类对 JSON 格式的数据解析，其中 JSONObject 用来解析“名称/值”对的集合，而 JSONArray 用来解析值的有序列表。

Android 终端的界面如图 5 所示，为了保持系统的一致性，该界面和 iOS 系统的界面类似，同样可以用大图标和页面底部的标签来进行相应的操作。

在图 6(a)中，我们可以看到，当我们从如图 5 所示的界面中点击“数字化进度”时，系统会通过调



Figure 4. iOS terminal interface

图 4. iOS 终端界面



Figure 5. Android terminal interface
图 5. Android 终端界面



Figure 6. Digital schedule of Android terminal
图 6. Android 终端数字化进度模块

用 REST 风格的 Web Service 服务器的 URL 资源地址下的接口 `public JSONArray getDigitalScheduleList(@FormParam("fondsCode") String fondsCode)` 和服务器进行交互，并获取 JSON 键值对数据 `{"serviceNo":"SZH2012060905","createDate":"2012-6-9"}`，该数据清晰地表示了正在进行数字化加工的文档的信息，即该文档的服务单号和创建时间，最后将这些数据显示在 Android 终端屏幕上。当我们点击这一条数据时，系统通过调用接口 `public JSONArray getDigitalScheduleByID(@FormParam("digitalID") String digitalNO)` 和点击该条数据时会获取的变量参数 `digitalNO` 得到 REST Web Service 服务器返回的 json 数据：`[{"digitalNo":"SZH2012060905","operatingDate":"2012-5-19","operatingProcess":"文档接收"}, {"digitalNo":"SZH2012060905","operatingDate":"2012-5-23","operatingProcess":"原件整理"}, {"digitalNo":"SZH2012060905","operatingDate":"2012-5-27","operatingProcess":"扫描"}, {"digitalNo":"SZH2012060905","operatingDate":"2012-6-15","operatingProcess":"图像处理"}]`，该 JSON 数组很明确的展

示了这一文档数字化加工的具体进度信息，即该文档的各个进度的不同时间，最后这些信息经过 Android 界面设计显示在屏幕上，如图 6(b)所示。

4.3. 小结

在本系统实例中详细介绍了 REST Web 服务的移动终端应用的系统架构，还采用了 JSON 作为数据交换的格式，并进行了实现，对实现的移动终端应用系统中的 REST Web 服务与前提的数据交互进行了详细的描述。

5. 结束语

基于 SOAP 型 Web 服务框架的性能开销大，RPC 复杂性高的缺点，而 REST 架构下的 Web 服务则是种轻量级的，且其实现和操作都比 SOAP 和 RPC 更为简洁，完全可以通过 HTTP 协议实现，还可以利用缓存 Cache 来提高响应速度，性能、效率和易用性上都优于 SOAP 和 RPC，而移动终端的存储容量和流量有限与移动互联网规模过快增加，通信数据也急剧增加，人们获取通过移动终端获取的信息量确很庞大现实相矛盾的实际情况，因此本文提出了 REST 风格 Web 服务在移动终端应用的框架，并且在 REST 框架中采用轻量级的数据交换格式 JSON 在不同平台间交换数据，更是有效地减小了带宽和流量。

本文详细介绍了 REST 风格 Web 服务在移动终端应用的框架，并且将其应用到实际的系统中，在两种主流的移动终端操作系统上实现了一个简化的文档管理系统，构建出了一个高效、松耦合、易访问的服务支撑系统，为传统的基于 PC 的系统移植到移动终端系统提供了一个范例。

参考文献 (References)

- [1] 冯新扬, 沈建京 (2010) REST 和 RPC: 两种 Web 服务架构风格比较分析. *小型微型计算机系统*, **7**, 1393-1395.
- [2] Fielding, R.T. (2000) Architectural styles and the design of network-based software architectures. University of California, Irvine.
- [3] Webber, J., Parastatidis, S. and Robinson, I. (2011) REST 实战. 李锲, 俞黎敏, 马钧, 等, 译. 东南大学出版社, 江苏.
- [4] Richardson, L. and Ruby, S. (2007) RESTful web services. O'Reilly Media.
- [5] 许卓明, 栗明, 董逸生 (2003) 基于 RPC 和基于 REST 的 Web 服务交互模型比较分析. *计算机工程*, **20**, 6-8.
- [6] Introducing JSON. <http://www.json.org>
- [7] Mark, D., Nutting, J. and LaMarche, J. (2012) iOS5 基础教程. 毛姝雯, 漆振, 杨越, 等, 译. 人民邮电出版社, 北京.
- [8] RestKit. <http://restkit.org/>
- [9] 林城 (2011) Android2.3 应用开发实战. 机械工业出版社, 北京.