

REST 在异构移动客户端接入中的应用研究

覃国蓉¹ 何涛²

¹(深圳市可视媒体处理与传输重点实验室 广东 深圳 518172)

²(深圳信息职业技术学院软件学院 广东 深圳 518172)

摘要 针对繁多异构的移动客户端接入 Web 应用系统的问题,提出一个基于 REST(Representational State Transfer)的解决方法并给出一个基于 MVC(Model-View-Controller)模式的具体的开发框架,通过 RESTful Web 服务连接移动客户端的 View/Controller 层与 Web 系统的 Model 层。定义了由 Web 系统的 model 层定义 RESTful Web 服务的具体方法和规则。理论分析与实际应用说明了该方法的有效性。

关键词 REST 移动互联网 客户端接入 异构移动客户端 Web 应用

中图分类号 TP3 **文献标识码** A **DOI**:10.3969/j.issn.1000-386x.2015.04.018

ON APPLYING REST TO HETEROGENEOUS MOBILE CLIENTS ACCESS

Qin Guorong¹ He Tao²

¹(Shenzhen Key Laboratory of Visual Media Processing and Transmission, Shenzhen 518172, Guangdong, China)

²(School of Software, Shenzhen Institute of Information Technology, Shenzhen 518172, Guangdong, China)

Abstract We propose a REST-based solution for the problem of various heterogeneous mobile clients accessing the Web applications and give an MVC pattern-based specific development framework, which connects the View/Controller layer of mobile clients and the Model layer of Web applications via RESTful Web services. We also give the definition that the specific method and rules of RESTful Web services are to be defined from the model layer of Web application. Theoretical analysis and practical application confirm the validity of the approach.

Keywords REST Mobile internet Client access Heterogeneous mobile clients Web application

0 引言

移动互联网的兴起和手机终端越来越强大,促使业界思考如何将业务系统从传统互联网模式转移到移动互联网上,并且能够借助手机终端强大的功能提供更方便的服务(如 LBS,基于位置的服务),现有海量的 Web 应用系统存在从 PC 客户端移植到移动设备的需求。但是移动设备和平台种类繁多^[1],所以研究如何将异构的移动客户端接入 Web 应用系统具有工程实际意义。

Web 应用系统通常采用模型-视图-控制器 MVC 模式实现。MVC 模式将系统分为 Model、View 和 Controller 三部分,为了能够集成各种异构移动终端,开发人员必须考虑如何连接移动客户端的 View/Controller 层与 Web 版的 Model 层。

REST 是 Fielding 在其 2000 年的论文中首次提出的一种软件架构^[2]。它定义了一个 Web 服务应用程序编程接口(API),REST 的交互直接使用 HTTP 协议,强调组件间的统一接口,从而信息能够以标准的格式来标示、定义和传输,极大地提高了系统的简单性和可扩展性。基于 REST 的服务叫做 RESTful Web 服务。相对于基于 WSDL 的 Big Web 服务,RESTful Web 服务是轻量级的 Web 服务。

本文在分析相关技术的基础上,提出了一个基于 REST 的

异构客户端接入方法。

1 相关研究与技术分析

目前虽然屏蔽异构系统的技术很多,但针对移动客户端接入问题的研究还比较缺乏。

屏蔽异构系统比较常用的框架和技术主要有分布式对象技术、多 Agent 系统和 Web 服务技术(包括 Big Web 服务和 RESTful Web 服务)。

文献[3]对包括分布式对象、Mobile Agent、Big Web 服务以及 RESTful Web 服务在内的技术进行了量化评估,得出了 RESTful Web 服务具有最好的可伸缩性、易用性、松耦合、可靠性以及可见性的结论。

同样是 Web 服务,文献[4]指出 RESTful Web 服务相比 Big Web 服务在扩展性、安全性、数据耦合性等方面存在着优势。

由于 RESTful Web 服务的客户端和服务端都免除了解析和封装 SOAP 数据包的性能消耗,也降低了传输的负载,交互性能高。文献[5]明确指出 RESTful Web 服务比 Big Web 服务更

收稿日期:2013-10-17。国家自然科学基金项目(61172165);广东省自然科学基金项目(S2011010006113, S2011040000672);深圳市科技研发资金项目(JC201006020791A);深圳信息学院科研项目(YB201008)。覃国蓉,教授,主研领域:软件体系结构,信息安全。何涛,讲师。

适合于移动应用环境。

在物联网应用领域已有部分研究者采用 REST 架构来构建应用平台^[6,7]。文献[6]提出 Web of Things 的架构,通过智能网关,将真实的物体转化为能通过 HTTP 直接访问的 REST 资源,整合到现有的 Web 系统中。文献[7]提出了一个基于 REST 架构风格的物联网服务平台,将传感节点等嵌入式设备和感知数据看作资源,使异构系统模块采用 REST 架构方式组建新型应用。

鉴于移动设备和平台种类繁多,移动设备的资源有限,将 Web 应用系统客户端从 PC 移植到移动设备上,对采用技术的系统实现的复杂性、扩展性和交互性能有较高的要求,在现有研究的基础上,受文献[6,7]启发,我们选择 RESTful Web 服务技术解决异构移动客户端的接入问题。

2 基于 REST 的异构移动终端接入方法

图1为采用 REST 接入异构移动客户端的开发框架。该框架基于 MVC 模式,通过 RESTful Web 服务连接移动客户端的 View 层和 Controller 层与 Web 系统的 Model 层。

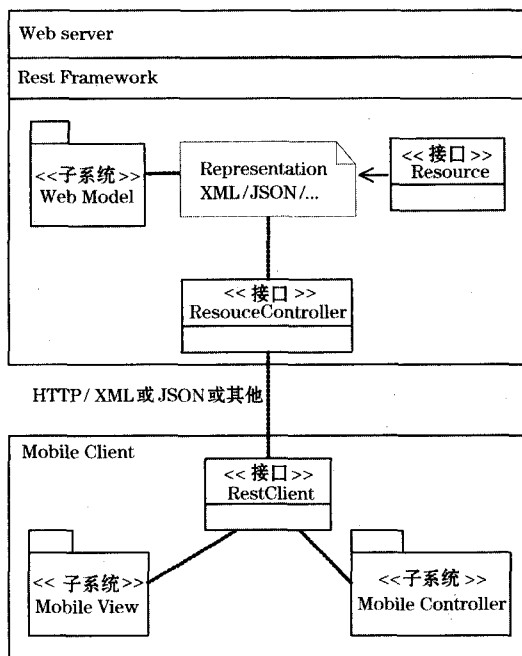


图1 采用 REST 接入异构移动客户端的开发框架

具体地,该开发框架包括以下 3 个步骤:

1) 首先,对 Web 版的 Model 层进行处理,定义统一的 RESTful Web 服务并发布服务接口,包括识别资源 Resource、定义资源表示 Representation、指定资源 URI 等步骤,其中资源 URI 就是 RESTful Web 服务的接口。如何根据 Model 层定义 RESTful Web 服务是本研究的重点和难点,将在第 3 节做详细介绍。

2) 然后,根据 RESTful Web 服务的技术选型实现 ResourceController 和 Resource 接口。ResourceController 接口规定了如何映射资源 URI 到 Resource 类;Resource 接口则规定 Resource 类如何调用 Model 类完成业务功能。如果选择 Restlet 2.0 框架,则可以通过继承 Restlet 2.0 的 Application 类自定义一个类实现 ResourceController 接口的功能,并通过继承 Restlet 2.0 的 ServerResource 类自定义一个 Resource 类实现 Resource 接口的功能。

在第 4 节会通过登录功能的实现介绍如何使用 Restlet 2.0 实现 ResourceController 和 Resource 接口。

3) 最后,根据 GUI 技术的选型,开发移动版的 View 和 Controller 模块并实现 RestClient 接口,通过 RestClient 请求对 Resource 进行操作。RestClient 接口规定了如何连接服务器端,实际上就是一个要求实现 HTTP 协议的客户端功能。如果是 Android 平台,直接使用 JDK 的 java.net 开发包或 Apache Common HttpClient 开发包都可以很容易地实现^[8]。

当服务器端接收到一个 RestClient 的请求时,ResourceController 会找到 Resource,通过 Resource 调用对应的 Model 类实现业务处理。如果需要增、删、改操作,则将请求负载中 Resource 的 Representation (JSON、XML 或其他) 也传递给 Model。如果是请求查询,则将查询结果转换成客户端需要的 Representation 格式作为响应负载返回给客户端。图 2 为处理移动客户端查询请求(查询 userid 为 qingr 的用户信息)的过程,数据格式采用 xml。

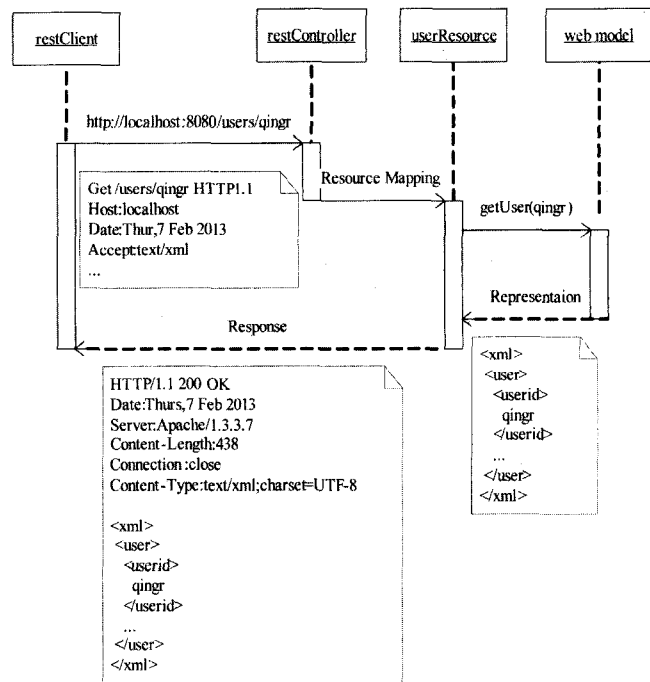


图2 移动客户端查询业务处理流程

3 基于 Model 层定义 RESTful Web 服务

基于 Model 层定义 RESTful Web 服务分为识别资源 Resource、定义资源的表示 Representation 和指定资源 URI 三个步骤。

3.1 识别资源 Resource

我们首先对 Model 层的各个类进行分析,将它们分为封装数据的 Entity 类和封装功能的 BO 类。基于 Entity 类形成初始的资源集合;将 BO 类的每一个功能,映射成增、删、查、改操作的资源,并对初始的资源集合进行调整。考虑到 Web 系统已经实现,所以以上 2 个步骤可保证识别的资源的完备性。

下面以企业信息系统最常见的表示用户的 Entity 类(假设类名为 User)和封装了登录功能的 BO 类(假设类名为 UserService)来说明如何识别资源。

分析 User,则可以得到表 1 所示初始资源集合。

表1 分析 User 类得到的初始资源列表

资源名称	HTTP 操作类型
User 的增加	PUT
User 的删除	DELETE
User 的修改	POST
对所有 User 的查询	GET
按主键(Userid)对 User 的查询	GET

UserService 通常通过按主键和密码查询 User 来实现登录功能,则需要增加一个资源:按主键(userid)和密码(password)查询 User,即通过分析登录功能调整后的资源集合为表2所示。

表2 分析登录功能修改后的资源列表

资源名称	HTTP 操作类型
User 的增加	PUT
User 的删除	DELETE
User 的修改	POST
对所有 User 的查询	GET
按主键(userid)对 User 的查询	GET
按主键(userid)和密码(password)对 User 的查询	GET

3.2 定义资源的表示 Representation

根据资源需要交换的信息来确定资源的表示 Representation。资源表示有多种数据格式,如 XML、JSON 等。如果是 XML 格式,“按主键(userid)查询 user”的返回结果的 Representation 如图3所示。对应的 JSON 格式如图4所示。

```
<user>
  <userid>qingr</userid>
  <password>123456</password>
  .....
</user>
```

图3 XML 格式 Representation

```
{"user":
  {
    "userid": "qingr",
    "password": "123456"
  }
}
```

图4 JSON 格式 Representation

3.3 指定资源 URI

我们采用如下命名规范:

1) 基于 Entity 类命名,将元素看作是集合的特例,如我们用/users 表示用户的集合,而/users/{userid} 则看做是给定 userid 值的某用户,当对应操作是 get 时,即返回给定 userid 值的用户信息,如果对应操作是 put/post/delete,则增加/修改/删除给定 userid 值的某个用户。这种方式可以最大化地统一资源命名并且具有良好的可读性。表3 为对表2 增加了 URI 的资源列表。

表3 增加了 URI 的资源列表

资源名称	HTTP 操作类型	URI
User 的增加	PUT	/users/{userid}
User 的删除	DELETE	/users/{userid}
User 的修改	POST	/users/{userid}
对所有 User 的查询	GET	/users
按主键(userid) 对 User 的查询	GET	/users/{userid}
按主键(userid)和密码 (password)对 user 的查询	GET	/users/{userid}/{password}

2) 由于一些 BO 类功能的实现涉及到多个 Entity 类,则用“-”表示关联,如学生选课涉及到学生类 Student 和课程类 Course,则学生选课的 URI 可以表示为/students-courses。

4 应用实例

“深圳市龙岗区义务工作者联合会义工综合信息管理平台”为 2011 年第 26 届大学生运动会配套服务系统,最初系统只是开发了浏览器版,后来用户又要求开发手机版以方便一些义工活动消息的及时推送。

为了快速生成不同平台(iOS、Anroid 和 Windows phone 等)的手机版系统,我们采用了基于 Rest 的接入方法。由于浏览器版的系统采用 SSH 框架完成而且已经上线正在运行,我们选择 Restlet2.0 作为 Rest 框架,这使得只需要修改 web.xml 增加简单的配置,即可集成手机版系统到原来的浏览器版,并且把对浏览器版系统的影响降到最小。考虑到各种编程语言均有 JSON 格式数据的序列化和反序列化支持,但 JSON 比 XML 要简便得多,我们的手机版系统选择 JSON 作为 Representation 的数据格式。

限于篇幅,这里只给出 Android 和 iphone(iOS)的登录功能的实现,其类图如图5所示。

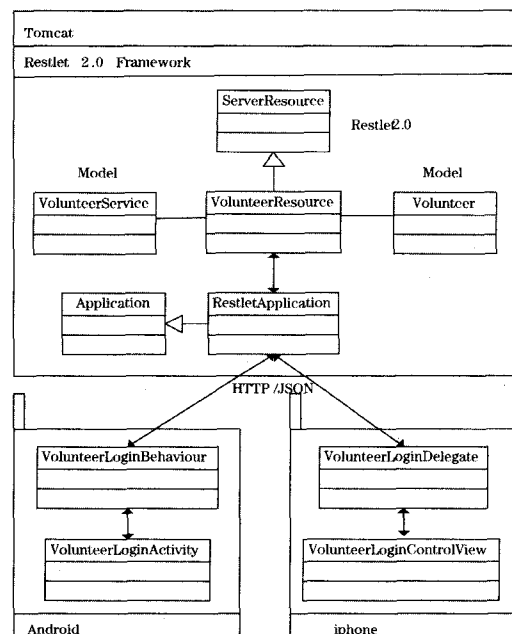


图5 Android 和 iphone 登录功能的实现

图中, Tomcat 为原来浏览器版的 Web 服务器; Volunteer 类 (下转第 144 页)

WLD 算法和本文算法在各种参数组合下的识别结果如图 9 所示,当 $M=6, T=8, S=4$, 分块数为 4×4 时,本文算法得到的识别率为 100%。当 $M=6, T=8, S=6$ 时, WLD 算法的识别率最高,即 92.2222%。实验结果表明,在三种分块情况下,本文算法的识别率都要高于 WLD 算法。

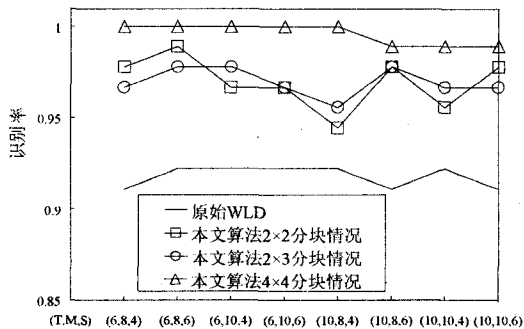


图9 本文算法和 WLD 算法的实验对比(YALE 人脸库)

本文也测试了 PCA 算法和 LBP 算法对 YALE 人脸库的识别率。图 10 给出了四种算法的最高识别率。可以看出,本文算法的识别率要优于其他几种算法。

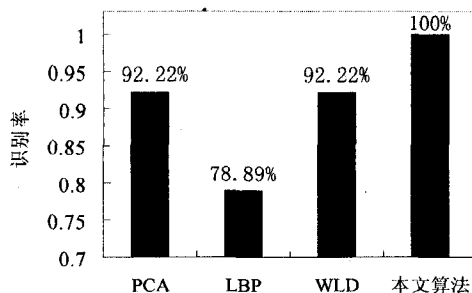


图10 不同算法在 YALE 人脸库上的最高识别率

4 结 语

本文对 WLD 特征的提取方法进行了改进,改进体现在两个方面:①对图像进行分块处理;②将梯度方向计算算子改为 Prewitt 算子。对本文算法的性能进行了实验验证,实验结果显示,在不同分块情况下,本文算法的识别性能都要优于 PCA、LBP、WLD 算法,从而表明了本文改进的 WLD 特征提取算法能够有效地提取人脸图像特征,对人脸表情、光照等变化具有很高的鲁棒性。

参 考 文 献

- [1] 曾彤. 基于嵌入式处理器指纹识别系统的设计和实现[J]. 电子设计工程, 2010, 18(4): 119-121.
- [2] 韦立庆, 陈秀红. 分块类推广 PCA 及其在人脸识别中的应用[J]. 计算机工程, 2011, 37(3): 166-168.
- [3] 谢永林. LDA 算法及其在人脸识别中的应用[J]. 计算机工程与应用, 2010, 46(19): 189-192.
- [4] 赵敏, 朱明. 基于 DTCWT 和 LBP 的低分辨率人脸识别[J]. 计算机工程, 2012, 38(22): 180-182.
- [5] Fan Xiao, Sun Yanfeng, Yin Baocai, et al. Gabor-based dynamic representation for human fatigue monitoring in facial image sequences[J]. Pattern Recognition Letters, 2010, 31(3): 234-243.
- [6] Ahonen T, Hadid A, Pietikainen M. Face description with local binary patterns: application to face recognition[J]. IEEE Transactions on Pat-

tern Analysis and Machine Intelligence, 2006, 28(12): 2037-2041.

- [7] Chen Jie, Shan Shiguang, He Chu, et al. WLD: A robust local image descriptor[J]. IEEE TPAMI, 2010, 32(9): 1705-1719.
- [8] 柳长源, 毕晓君, 韦琦. 基于向量机器学习算法的多模式分类器的研究及改进[J]. 电机与控制学报, 2013, 17(1): 115-117.
- [9] Luo R C, Lin T T, Chen Kuanyu. Gender recognition based on ensemble learning with selective features for service robotics applications[C]. IEEE International conference on robotics and biomimetics, 2011: 1159-1164.
- [10] 姜炳旭, 刘杰, 孙河. Sobel 边缘检测的细化[J]. 沈阳师范大学学报: 自然科学版, 2010(4): 503-505.

(上接第 79 页)

和 VolunteerService 分别复用浏览器版的 model 层的类; 而 RestletApplication 继承 org.restlet.Application 类完成 RestController 的功能——匹配相关 URI 和 Resource 类; VolunteerResource 则继承 org.restlet.resource.ServerResource 类完成 Resource 接口的功能, 该类调用 model 层的 VolunteerService 和 Volunteer 类查询数据库, 实现身份认证。

对于 Android 移动终端, VolunteerLoginActivity 实现界面显示, 其界面事件处理委托给 VolunteerLoginBehaviour 处理, 包括 RestClient 接口的功能: 连接到 Web 服务器。

对于 iPhone 手机, VolunteerLoginControlView 实现界面显示, VolunteerLoginDelegate 处理事件, 包括 RestClient 接口的功能。

5 结 语

针对当前移动互联网迅速发展的背景, 本文提出了一个基于 REST 的异构移动客户端接入方法, 理论分析和实际应用验证了该方法的有效性。不难证明该方法还可以用于实现 PC 桌面客户端。

参 考 文 献

- [1] Andreas Holzinger, Peter Treitler, Wolfgang Slany. Making Apps Useable on Multiple Different Mobile Platforms: On Interoperability for Business Application Development on Smartphones[C]//2012 International Cross-Domain Conference and Workshop on Availability, Reliability, and Security. Heidelberg: Springer Verlag, 2012: 176-189.
- [2] Roy Thomas Fielding. Architectural styles and the design of network-based software architectures[D]. Doctoral Dissertation, Dept. of Computer Science, Univ. of California, Irvine. 2000.
- [3] 王姜, 余萍, 曹春, 等. 开放网络环境下的程序设计: 从 RPC 到 REST[J/OL]. 计算机工程与应用, 2012-11-30. <http://www.cnki.net/kcms/detail/11.2127.TP.20121130.1126.004.html>.
- [4] 冯新扬, 沈建京. REST 和 RPC: 两种 Web 服务架构风格比较分析[J]. 小型微型计算机系统, 2010, 31(7): 1393-1395.
- [5] Feda AlShahwan, Klaus Moessner. Providing SOAP Web Services and RESTful Web Services from Mobile Hosts[C]//5th International Conference on Internet and Web Applications and Services. Piscataway: IEEE Computer Society, 2010: 174-179.
- [6] Guinard D, Trifa V, Pham T, et al. Towards physical mashups in the Web of things[C]//INSS'09, 2009.
- [7] 程冬梅, 王瑞聪, 刘燕, 等. 基于 REST 架构风格的物联网服务平台研发[J]. 计算机工程与应用, 2012, 48(14): 74-78, 157.
- [8] Jose standoval. RESTful Java Web Services[M]. Birmingham: Packt Publishing Ltd. 2009.