

RESTful 风格的泛在学习平台的设计和实现 *

尚飞, 张勳

(北京邮电大学网络教育学院, 北京 100876)

摘 要: 通过研究 Rails 框架实现 RESTful Web Service 的原理, 本文从资源规划和 URI 设计, 路由建立, 控制器设计以及实现模板等方面综合阐述了泛在学习平台的设计思路, 利用 MVC 架构和 Ruby 语言实现了具有良好交互性和可伸缩性的泛在学习平台, 使用户可以轻松地使用平台进行学习和交流。

关键词: 泛在学习; Web 应用; RESTful; Rails

中图分类号: TP311.52

文献标识码: A

DOI: 10.3969/j.issn.1003-6970.2013.11.001

本文著录格式: [1] 尚飞, 张勳. RESTful 风格的泛在学习平台的设计和实现 [J]. 软件, 2013, 34 (11): 1-4

Design and Implementation of Ubiquitous Learning Platform based on RESTful

SHANG Fei, ZHANG Xu

(School of Network Education, Beijing University of Posts and Telecommunications, Beijing 100876, China)

【Abstract】 This article describes the design of the ubiquitous learning platform from resource planning & URI, route establishment, controller design and template implementing and other aspects based on the Rails framework to achieve RESTful Web Service. A good interactive and scalable ubiquitous learning platform is implemented by using MVC architecture and Ruby language, so that users may learn and communicate easily on this platform.

【Key words】 Ubiquitous Learning; Web Application; RESTful; Rails

0 引言

泛在学习作为一种新型的学习模式倡导一种无时无刻、随心所欲的学习理念, 客观世界中的事物实体或者社会环境都可以成为被认知的对象, 融合移动数字化学习的优点, 用户使用移动终端接入 Web 应用服务系统或者嵌入式交互设备来支持学习, 实现以学习者为中心, 学习内容为重点的泛在空间。

传统上实现这样的泛在学习平台是基于用户的动作来设计的, 现有多数框架能够支持这种设计方式, 常见的框架是以 SOAP 协议为信息传递的基础, 借助 HTTP 协议的传输功能, 采用面向动作的方式, 完成信息的交互, 其缺点是复杂的嵌套协议增加了开发的难度, 自定义的方法使得不同系统间产生了交互阻碍, HTTP 协议仅作为庞大 SOAP 负载的传输协议。

RESTful 作为分布式超媒体系统设计的一种框架风格, 一般是指使用 HTTP 协议的语法和语义将功能完全转化为一组 URI 可寻址资源来向 Web 应用的服务对象提供 Web 支持的架构方式, 它将资源和资源表现分离, 为构建下一代简单性、高效能, 易移植的 Web 应用提供了若干基本准则, 使得开发效率和程序性能大大提升, 非常适合用于快速开发泛在学习平台。

1 国内外研究现状

目前国外关于泛在学习的实践项目比较知名的包括: 英国的环境森林计划 (Ambient Wood Project), 该项目将计算机放置在森林中, 采集森林中各类信息, 用户无需亲自前往森林也能够即时地获得森林中最新的信息; 日本开发的基于情景的语言支持系统 (JAM IOLAS), 通过时刻感知外界环境, 根据分析数据向用户提问, 从而帮助日语学习者掌握如何使用日语拟态词和拟声词来描述身处的环境。

同时基于 RESTful 风格的各类 Web 应用服务也得到了广泛的研究和应用, 国外比较著名的应用包括 Twitter、Google 提供的 API 以及项目管理工具 Redmine 和 Github, 国内使用 RESTful Web Service 的平台研究涉及信息共享与整合平台, 瓦片式地图服务和学术资源发布平台等。

2 泛在学习平台关键技术

2.1 RESTful 与 Rails

2.1.1 RESTful Web Service 的设计标准

RESTful Web Service 以资源为中心, 任何人们可以认知的事

* 基金项目: 国家科技重大专项课题 (专项名称: 新一代宽带无线移动通信网, 课题名称: 基于 Web 的无线泛在业务环境体系架构、关键技术研究及演示验证, 编号: 2012ZX03005008)

作者简介: 尚飞 (1987—), 男, 硕士研究生, 研究方向: 通信与信息系统。

导师: 张勳, 女, 副教授, 研究方向: 信息与通信工程。

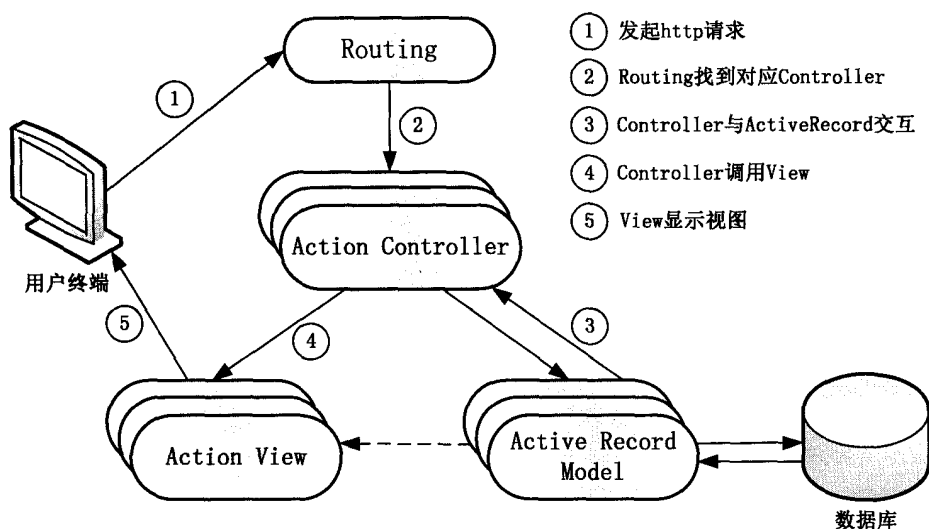


图 2.1 Rails 的 MVC 框架

物,只要具有被引用的必要,它就被描述成资源。以资源为核心的 RESTful 对 Web 应用的设计标准如下:

- (1) Web 上的所有事物都被统一抽象为资源 (resource);
 - (2) 每一个资源都有一个唯一的资源标识符 (resource identifier);
 - (3) 使用通用连接接口操作资源;
 - (4) 资源标识符是静态的,不会随着资源的具体操作变化;
 - (5) 所有的操作都是无状态的 (stateless);
- 满足以上标准的 Web 应用具备以下几个优势:

(1) 简明 URI: URI 代表资源,由控制器名称和资源 id 组成,不包含对资源的动作;

(2) 统一操作接口:任何对资源的操作都是通过独立于资源的统一接口实现的,使用 HTTP GET 获取资源;使用 HTTP POST 创建新资源;使用 HTTP PUT 修改资源;使用 HTTP DELETE 删除资源;

(3) 资源具备多种格式:controller 的一个方法,可以返回不同的数据格式,包括 XML, HTML, JSON 或 RSS 等;

(4) 面向 CRUD (Create、Read、Update、Delete) 的控制器:每一个控制器都具备这四种对资源的基本操作,在此基础上用户可以扩展自定义的操作;

(5) 更少的代码量和更简单的设计;

2.1.2 Rails 框架

Rails 是动态语言 Ruby 支持下的全栈、开源的敏捷 Web 开发框架,包括自带 Web 服务器、处理 HTTP 请求和响应以及储存数据到关系数据库,完整实现了“Model-View-Controller (MVC)”框架,如图 2.1,Active Record 通过封装数据库以及数据库访问过程来实现 Model,其中基类 Active Record::Base 定义 CRUD 操作,用户自定义方法都是继承自基类;Action

Controller 组织并处理来自数据库和用户发送的数据,然后一并提交给 Action View,最后由 Action View 将数据载入模板显示;Action View 是基于嵌入式 Ruby (ERb) 的视图系统,通过视图模板如 .html.erb 等文件定义数据表现形式来实现 View。

2.2.3 Rails 实现 RESTful 的原理

Rails 框架使用 URI 来标识资源,利用控制器来访问资源,一个控制器包含支持一个 Web 服务的若干方法,Rails 中资源的 URI 由控制器名和模型类名组成,不包括动作,但是可以将 HTTP 的四个动词与 URI 组合形

成对资源的操作,在框架的 config/routes.rb 中设置每个资源模型的操作集,每一个操作包括 HTTP 动词,资源 URI,控制器方法和路由的宏定义,Rails 根据 routes.rb 将用户请求路由到指定控制器的方法,通过控制器的进一步组织和处理最终明确 Rails 把结果递交给哪个类以及以何种数据格式表现,Rails 通过 HTTP 请求头部和 URI 来传递所需资源的表现形式,只要指定浏览器发送的 HTTP 请求头部的 Accept 参数,服务器就可以据此决定返回的数据格式。Rails 就是实现了将 URI 转换为控制器可以处理的对象,然后用控制器转换成资源,资源进而转换为 Active Record 对象,最后把 Active Record 对象转换为数据库记录。

2.2.4 使用 Rails 的优势

(1) 与生俱来的易扩展性

Ruby 社区有很多成熟的 Gem, Gem 是使用 Ruby 编写的解决某一类开发需求的开源工具包,在工程目录下执行“gem install 包名称”就可以方便地整合并使用某个工具包,如此可以大大降低开发成本,相比刚刚兴起的 PHP 的包管理 Composer, Rails 已经具备众多高质量的 Gem。

(2) 开发效率很高

开发者不需要注重底层代码构建,因为 Rails 提供的基础方法很多,另外 Rails 3.2 中还整合了 23 个常用的 Gem,这都极大地提高了开发效率,相比之下 PHP 或者 Java 仍要从头构建很多底层的代码。

(3) 数据库可移植性强

Rails 的数据库设定档是 config/database.yml,其中包含三段不同环境的设定,对应三个 Rails 执行环境 (development, test, production),每个执行环境都可以设定独立的数据库配置,如此可以应对多种数据库,便于移植。

(4) 简单的环境配置

假如项目开发过程中需要更换或者添加新硬件, 只需保持新硬件上安装版本一致的 Ruby 和 Rails, 然后在工程目录下执行“bundle install”, 就可以把移植的项目所需的环境配置好, 完全不必依照项目开发手册逐一安装所需配置, 相比之下 PHP 的非标准库, 不仅需要 phpize 编译, 而且还要修改 php.ini, 比较复杂。

(5) 低学习代价

PHP 和 Java 的学习代价略高于 Rails, 不适合快速低成本开发。

3 泛在学习平台的设计和实现

3.1 平台总体结构设计

3.1.1 目标和原则

首先, 泛在学习平台是围绕课堂日常教学任务设计的, 应当包括最基本的几类课堂活动, 如安排课程, 授课通知, 编辑课程资料, 随堂测验以及布置课后习题等传统的教学活动。

其次, 泛在学习平台作为增强师生之间教学互动的新途径, 要充分利用数字化网络的优势, 开展一系列新颖的线上线下教学活动, 根据活动的反馈结果利用第三方 API 即时动态地分析教学效果。

最后, 构建一个支持线上课堂教学与线下学习互动的泛在学习平台, 使学生可以不局限于课堂这种传统的教学形式, 体验数字化, 网络化的教学带来的便利, 同时也可以作为高校教学的一种辅助手段。

3.1.2 总体结构

本平台由浏览器应用层、业务层, 服务层和数据访问层自上而下组成, 如图 3.1。浏览器应用层是具备 Web 浏览器的用户终端, 负责接收用户请求和显示资源视图; 业务层是用户调用服务层的接口, 负责处理应用层的 HTTP 请求, 根据 routing 找到预设的控制器方法, 然后将请求路由到服务层某个模块; 服务层是平台的核心, 以 RESTful 的资源访问方式来实现各类服务, 所有的被引用的资源都是在服务层以指定的控制器方法完成由 URI 向用户需求或服务提供的格式转换、组合和更新, 包括实时问题墙, 在线问答, 在线投票, 群组讨论以及课堂测验等服务组件; 数据访问层定义了各类型数据库的操作, 封装了操作细节, 对上层提供数据访问的接口。

3.2 平台的实现

实现泛在学习平台服务层的功能主要分成四个部分, 一需要根据需求建立资源模型, 主要是在框架内 app/models/ 目录下建立 .rb 文件, 编写资源模型的两类声明; 二是分析基于资源的操作、逻辑和服务, 设计符合 RESTful 标准、一层次清晰的 URI, 其中需要包含动态参数和控制说明, 最后为每一个 URI 指定标准 HTTP 动词和控制器方法, 形成完整的资源操作定义; 三

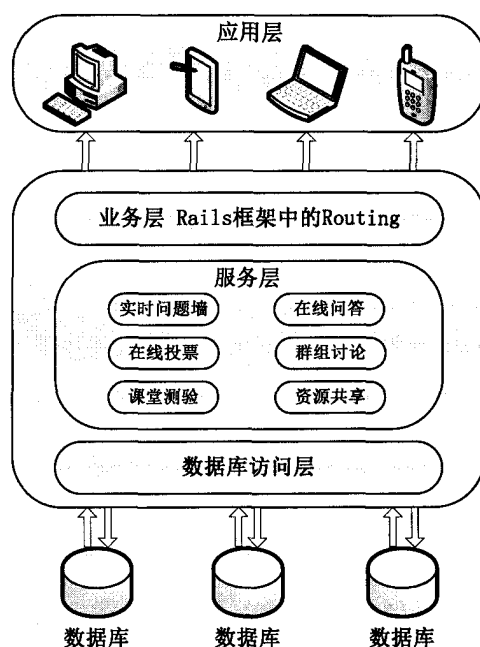


图 3.1 平台架构

是编写控制器的方法, 对于简单的面向资源的 CRUD 操作可以使用 scaffold 来直接自动生成, 对于自定义的控制器方法, 需要遵照 config/routes.rb 中的声明, 实现方法内部逻辑并返回多种数据格式; 四解析控制器方法返回的数据, 结合 html、xml 等语言组成视图, 设计符合用户需求, 具备良好交互性的复用的视图模板。

实现服务层每个功能模块的主要步骤:

第一步: 在 app/models/ 目录下生成 .rb 文件, 首先编写模型间映射关系的声明, 包括 belongs_to(属于), has_many(一对多), has_one(一对一), 形如 belongs_to: questions; 然后声明数据库中对对应表的域, 形如 attr_accessible: id, name, age, address, 必要时可以使用 validates 指定校验域和校验方式。

第二步: 在 config/routes.rb 文件中结合关键字 resources 和 collection 设计资源 URI, 指定 HTTP 动词, 关联控制器方法以及路由的宏定义, 形如 get "/quizzes/:quiz_id/quiz_start" => 'answer#start', :as => "quiz_start" 的 URI, 其中 get 表示使用 HTTP GET 方法来操作资源, "/quizzes/:quiz_id/quiz_start" 由三层结构组成, 访问此 URI 会关联 "answer#start", 其中 :quiz_id 是动态参数, 由 answer 控制器的 start 方法生成, 关键字 as 表示可以使用 quiz_start_path 来表示 "/quizzes/:quiz_id/quiz_start", 相当于程序化语言中的宏定义。

第三步, 编写控制器的方法, 简单地使用 scaffold 命令自动生成 CRUD 基本操作, 而自定义的操作需要在 app/controllers 目录下对应 .rb 文件中添加具体代码, 常用变量和语句有 params 变量, session 变量, new 方法, save 方法, each.do 循环语句, respond_to 语句以及 redirect_to 语句等。

第四步, 由于控制器方法可以利用 `respond_to` 语句返回多种数据模板 (`format.html, format.json, format.xml` 等), 所以需要在 `app/views` 目录下对应文件夹内添加若干视图模板用于响应控制器的各种返回结果, 一般视图模板在 “<%” 和 “%>” 之内嵌 ruby 语句处理控制器中的 Ruby 变量, 如以 @ 开头的变量, `params` 变量和 `session` 变量。

以上四步是实现问题墙, 在线问答, 在线投票, 群组讨论, 课堂测验和资源共享的通用流程, 根据功能模块

的迭代开发原则, 需要不断重复第三步和第四步完成模块扩展。

3.3 使用第三方 API 增强平台功能

在线投票模块引入了谷歌提供的在线绘图 API: Google Chart API 即时动态地分析投票结果; 随堂测验模块同样引入该 API 将学生的测验结果以多样化的图表方式分析后呈现给教师。增强后效果图如图 3.3 和 3.4。

4 结束语

泛在学习平台是融合泛在学习与 Web 应用的一次有力探索, 验证了平台开发的可行性并利用合适的技术手段尝试实现平台基本功能。开发过程中遵循 RESTful Web Service 的设计标准, 在使用 Rails 框架简明快速地实现传统课堂教学任务的基础上, 通过引入第三方 API 的方式增强平台功能, 即时动态地分析教学数据, 为用户之间形成良好的互动提供依据, 为进一步与同类第三方 API 整合提供经验。本平台在后台编码、模块复用和响应速度方面尚有待解决的问题, 此问题的解决可以使得此平台具有更大的应用空间, 不仅能够形成新型的教学模式, 而且可以成为更加广泛的融合学习娱乐的社交平台。

参考文献

- [1] Roy Thomas Fielding. Architectural Styles and the Design of Network Based on Software Architectures [D], UNIVERSITY OF

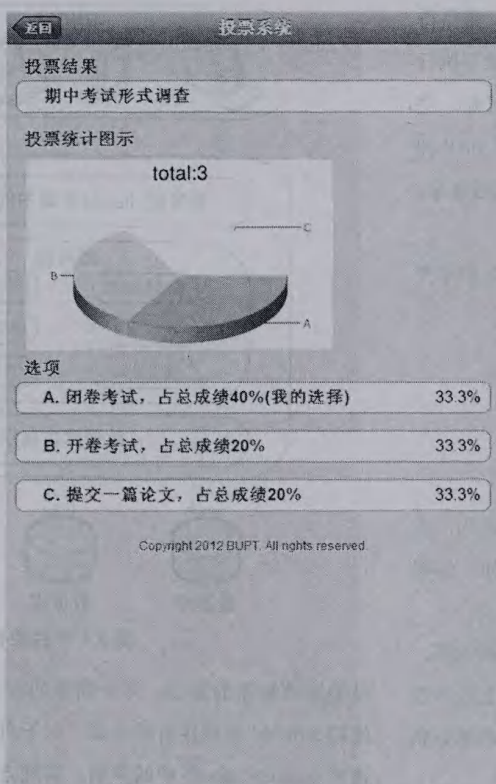


图 3.3 投票结果

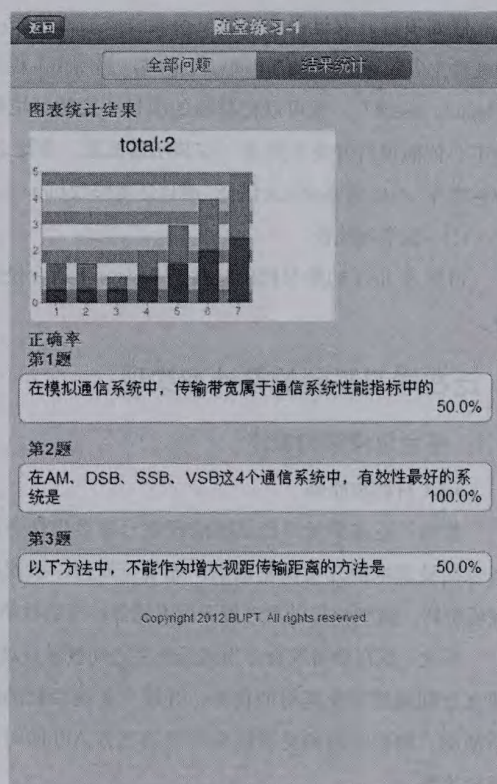


图 3.4 测验结果

CALIFORNIA, 2008: 88.

- [2] Nan Feng, Jing Xie, Yingxin Wu. Comparison of Ruby on Rails Development Tools. Software Engineering, 2009: 290 - 294
- [3] Perera, S., Herath, C., Ekanayake, J.. Axis2, Middleware for Next Generation Web Services. Web Services, 2006: 833 - 840
- [4] Yen-Min Huang, Ravishankar, C.V. Designing an agent synthesis system for cross-RPC communication. Software Engineering, IEEE Transactions, Mar 1994: 188 - 198
- [5] Gao Linqi, Li Congdong. Hybrid Personalized Recommended Model Based on Genetic Algorithm. Wireless Communications, Networking and Mobile Computing, 2008. 1 - 4
- [6] Perera, S., Herath, C., Ekanayake, J.. Axis2, Middleware for Next Generation Web Services. Web Services, 2006: 833 - 840
- [7] 董京峰. 泛在学习—学习者自由探索的新方式 [J]. 计算机时代, 2010, (5): 56-57, 59. DOI: 10.3969/j.issn.1006-8228.2010.05.023.
- [8] 周海涛. 泛在网络的技术、应用与发展 [J]. 电信科学, 2009, 25(8): 97-100. DOI: 10.3969/j.issn.1000-0801.2009.08.021.
- [9] 于晶, 蒋挺. 无线传感器网络移动锚点定位技术综述 [J]. 新型工业化, 2011, (12): 7-11.
- [10] 移动泛在业务环境及其体系架构设计的挑战 [J]. 北京邮电大学学报, 2005, 28(5): 1-3, 37.
- [11] 王建斌, 胡小生, 李康君等. REST 风格和基于 SOAP 的 Web Services 的比较与结合 [J]. 计算机应用与软件, 2010, 27(9): 297-300. DOI: 10.3969/j.issn.1000-386X.2010.09.093.