

论基于 REST 服务的 Web 应用系统设计

【摘要】

2011年上半年，我在上海中软资源软件有限公司(ICSS)，作为项目组长参与了公司人事管理(HR)系统开发。在系统开发前，公司在信息化建设中，业已采用请假流程、薪资管理、招聘等系统，虽然较为成熟，但彼此间互相独立，业务数据无法共享。且公司各个分公司间，对 HR 系统使用情况也截然不同，有的分公司由于各种原因，仍然采用手工管理本应信息系统的业务流程。公司是以软件外包业务为主，所以人力资源管理系统在公司信息化建设中的地位至关重要。这次开发的 HR 系统，将整合现有的业务系统，在整个公司内部推行使用，以解决信息孤岛带来的效率低下问题。为了以后的扩展需要，保证在业务和空间尽可能大的扩展性。因此，经过研讨，决定采用 REST Web 服务方式实现系统应用层。本文将就 HR 系统开发过程，描述一下对 REST 服务的使用和认识的体会。

【正文】

上海中软 HR 管理系统整体采用基于 B/S 的三层架构设计。

我做为项目组长参与系统需求分析至测试和部署的整个过程，直接向 IT 部门总监汇报。负责沟通需求，建立项目组，确定系统架构风格和技术实现方案。预定开发周期为 120 天，系统部署后有两个月的试运行期，项目组人数在 5-10 人间变动。由于项目开发资源（比如时间）紧张，公司 HR 系统业务逻辑复杂，旧系统改进与新需求交织，项目组对业务并不熟悉，难以在一开始预估将所有业务移植到新系统的时间。因此，在开发模型选择上，采用螺旋式增量开发。首先不必追求大而全，在开发完系统基本框架基础上，优先移植最亟待改进的业务。经与领导和 HR 部门沟通研究，递交了系统准备实现的功能列表，按不同实现优先级排列，标记为 P1 的功能优先级最高，必须实现。标记为 P2/P3/P4 的功能优先级依次降低，必要时可以根据资源情况需要进行裁剪。

在开发技术的选择上，由于本公司业务以微软外包为主，公司的开发人员大都熟悉一项或多项微软开发技术，作为微软公司合作伙伴可以低成本获取软件开发和管理工具，方便地获取技术支持。所以决定该系统采用微软技术：表示层基于 ASP.NET 4.0；中间业务层采用 REST 服务实现，基于 WCF(Windows Communication Foundation) 4.0；数据访问层基于微软的 ORM 构件-AEF(ADO.Net Entity Framework) 4.0。在构件的选择上，尽可能降低开发工作量，提高效率，力求避免把主要精力放在通用的技术细节，而是放在业务逻辑的研究和实现上。

系统部署共有三台服务器：两台 Web 服务器 Windows Server 2008 + IIS 7.5，分别运行系统网站及 REST 服务；一台数据库服务器 Windows Server 2008 + SQL Server 2008。经过试运行，于 7 月份投入正式使用。目前系统状况良好，经运行评估，实现了全部必须功能，性能、安全性等质量均达到了原定设计要求。目前系统正在根据业务需要，由后续项目组做二次开发中。

采用 REST 服务方式实现系统业务逻辑层，完全符合项目开发时考虑的两个因素：简单和灵活。传统的 Internet Web 服务一般基于 SOAP 协议，构造 SOAP 请求 XML 虽然目前.NET Framework 已实现较好地封装，但不便非.Net 语言调用，如客户端页面中大量采用了 Ajax 技术，使用 JavaScript 构造 Soap 请求非常困难。在调用服务的 Web 页面开发完成前，为了调试和测试服务，必须写单独的测试程序，十分不便。

相比之下，而 REST 服务具有非常出色地灵活性。既能被服务器端面向对象语言调用，又可以直接被客户端的脚本语言调用。也很方便用浏览器和 Fiddler 工具进行测试。我们在项目中，并没有将 REST 服务单纯视为一串地址的响应，但基于 HTTP 协议，可以最大地利用 HTTP 协议的语义特性。如数据的增删改查操作对应不同 Http Method(Put/Delete/Update/Get)。用户可以用相同访问服务结点(Endpoint)，根据需要，通过在请求头中设置不同的 Accept-Type，获取不同形式的数据结果，比如 JSON (用于 Ajax)或 XML(用于后台)。

更好的性能和缓存支持——由于不需要构造 Soap 消息，请求 Rest 服务显然开销更小。REST 类 Web 服务可以利用高速缓存控制头，从而减少带宽的需求，从而 REST 可以改善响应时间和改进用户体验。

可扩展性和无状态性——每个请求都是独立的。一旦被调用，服务器不保留任何会话，这样就可以更具响应性。通过减少事件后通讯状态的维护工作，提高了服务器的可扩展性。

在为系统开发 REST 服务时，也遇到一些问题：

一、安全性方案。并不是指 REST 服务安全性不足，其本身没有内置的安全支持，但所有 HTTP 支持安全模式和框架几乎都可以用于 REST 服务。真正潜在风险存在于 REST 灵活的使用方式上，既可以被服务器端调用又能被客户端调用，所以一开始就要明确地区分用户访问权限和系统访问权限，区分 Web 页面权限和 REST 服务权限，但有时在开发中经常混为一谈，所以要加强设计阶段这方面的文档和评估工作。

二、服务接口规范性。REST 服务基于 URI 地址访问，有非常强的语义性，服务接口的每个操作都基于一个 URI 模板。在实际业务中，功能类似的操作被做成多个重载，随之重载的增多，URI 模板如何约定，如何扩展便成为一个规范性问题。开始时，对此未予以足够重视，在多人开发服务，以致一些服务操作语义产生了混乱，影响了理解和正确使用。后来，又额外花费时间资源统一了规定了操作 Uri 格式。这一方面，源于业内尚无明确的标准，更重要是，应该从设计时就全面考虑将来如果需要重载等功能扩展，URI 模板的语义扩展方式。还有一些其他的规范问题，诸如一些操作包括增删改查中的一种以上的数据操作，Http Method 如何定义，也应该一并考虑。

三、WCF REST 自身限制。WCF 从 3.0 发展到 4.0，已经是较为成熟。而 WCF 的 REST 构件，则是全新的技术，WCF 作为.NET 平台 Web Service 的替代者，无论在开发还是管理上，都极大的灵活性。而 WCF REST 的灵活体现在开发和使用上，在管理维护情况下，WCF REST 服务接口操作未提供如 WCF 一样的灵活的配置功能，URI 模板等元素必须在代码中设置，消息格式虽然可以根据客户端请求输出，但不能在配置文件中设置。

总的来说，虽然 REST 服务仍然在发展中，经验与技术还有很大进步空间。但毫无疑问，基于 REST 服务的 WEB 应用程序拥有很多优势，未在在 WEB 系统，将有更光明的应用前景。

【原文链接：https://blog.csdn.net/weixin_34077371/article/details/86469807】