

RIA REST如何来化解Java的劣势 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/237/2021_2022_RIA_REST_E5_A6_c104_237351.htm Java的劣势在何处？与前些年相比，现在看的已经很清楚，Java的劣势就在于做Web表现层的开发。Web表现层开发需求变化频繁，Java这类静态类型的语言不够敏捷，严重影响了开发的效率。而JavaEE的一个最大的缺点，就是企图在服务器端搞定一切，我将这种开发方式称作“传统集中式的开发方式”。标准的J2EE三层架构Web表现层、业务层、持久层，也许对于传统的基于HTML表单的Web应用来说是恰当的，但是现在已经显得落伍了。JavaEE企图在服务器端完全搞定Web表现层的开发，给自己制造了一个大麻烦。无论是从这门语言本身，还是从支持这门语言主要的公司Sun、IBM、BEA、Oracle来说，他们并不擅长此道。擅长此道的是哪些公司呢？Adobe/Macromedia、M\$、Borland/CodeGear。如果Web表现层必须要在服务器端开发，Ruby on Rails的优势与JavaEE相比要明显的多。RoR要比任何主流的JavaEE Web表现层框架和技术（Struts、WebWork、Spring MVC、JSF、Tapestry、etc.）更加灵活，学习成本更低，开发效率更高。换个思路来思考，如果我们不再假设客户端就是几乎毫无智能的Thin Client将会如何？假设我们能够充分利用客户端的Ajax组件库和各种RIA技术，将Web表现层完全或者绝大部分前推到客户端来开发，并且通过REST风格的API来与服务器通信，那么服务器的角色就变成了类似于Web服务提供者（注意：这里和Web服务还是有很大的差别，因为REST在这里是用于同一个应用内部的通信，即连接

一个应用的客户端和服务端)的角色,这样就能够极大地简化服务器端Java的开发工作,让它从自己所不擅长的领域退出来,集中精力做自己最擅长的一些工作。这个趋势其实在3年多前我在JavaEye论坛中宣传基于XMLHttpRequest的开发方式的时候就已经看到了,现在这个趋势已经越来越明显了,新一代Web开发方式的面貌已经逐渐浮出水面。Adobe AIR/Flex、M\$ WPF/Silverlight都是这样一类的开发方式,当然Ajax也可以以这种方式来做开发。我给这样一类开发方式取名叫做“RIA REST”。在服务器端搞定一切当然也有好处,因为这样可以获得最佳的控制,安全问题解决起来也比较容易。但是其代价就是无法得到最佳的交互设计,强迫用户不得不承受降级的使用体验。如果这样的用户体验是能够接受的,那么采用这种方式做设计和开发问题不大。但是如果这样的用户体验是无法接受的,那么就需要严肃地考虑RIA REST的开发方式了。与传统集中式的开发方式相比,这是一类新型的分布式的开发方式,在一些方面(交互设计、服务器端架构)得到了简化的同时,也会使得一些方面(服务器端的控制能力、安全性)复杂化,所以要求架构师作出慎重的权衡。分布式应用必然会带来很大的复杂性,但是REST无疑是基于Web的分布式应用的最理想的架构风格,在Web领域REST的优势要比RPC和分布式对象等架构风格大的多。同时REST是简练实用的,可以很大程度上降低分布式应用的巨大复杂性。

100Test 下载频道开通,各类考试题目直接下载。
详细请访问 www.100test.com