

RIA给Java在Web开发领域带来转机 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/245/2021\\_2022\\_RIA\\_E7\\_BB\\_99Java\\_c104\\_245768.htm](https://www.100test.com/kao_ti2020/245/2021_2022_RIA_E7_BB_99Java_c104_245768.htm) 如果采用RIA开发网站，和传统web开发有两点很大的区别，一是：对于服务端View层的依赖就没有了，二是：不需要在server端保存session信息。无论是rails，还是php，python，Java的各种web框架，session和server view都是必须的，从上个世纪中期到现在10年多的时间里，web的发展和http本身无状态的特点使得我们无法脱离这两点来开发网站，几乎所有web框架都会涉及到它们，但是都会采用各自的方法来解决，每个方案都没有本质的不同，却在形式上差别很大，服务端view技术可能是每个框架中最麻烦的部分。REST本身并没有改变这种开发现状，但是RIA却在试图打破这个局面。一旦采用了RIA，你会发现，各种框架其实在编写业务逻辑和服务方面(ORM层，Router)并没有什么显著的不同，单从Controller和Model方面考虑，脚本语言web框架还是有一定的敏捷优势的，但是Java框架已经和它们相差不多了，但是Java在纵向扩展性，执行效率，静态检查，组件等很多方面具备脚本语言所没有的优势。所以如果一旦未来的Java web框架简化到一定程度，Java web开发的优势就会显现出来，所以我个人感觉RIA的出现对于Java可能是一个转机。当然这要取决RIA的接受程度。GWT就是一个好的开端，不过我更喜欢基于flash的解决方案。成熟高效的runtime，flash插件庞大的市场占有率，跨平台，跨浏览器，Actionscript已经发展到第三版，优良的组件编程，多媒体的支持。所有这些似乎都预示着flash RIA有着良好的前程，但是事实也不是这样的。

对于一种新技术的采纳，并不是那么简单。更多的人是在观望，flash RIA有学习成本，熟悉的人不多，资料也少；搜索引擎优化也是一个问题，而且在开发效率上也不见的高(这的确需要实践一下)。最近我打算开始研究一下flex，以前我总觉得flex是Macromedia(adobe)兜售的一种服务端技术，自从adobe推出了apollo后，我觉得flex变成了一个开发客户端应用的平台。<http://pownce.com/>网站上那个apollo air的小应用，让我看到了学习flex的必要。本来flex是一个服务端应用技术，现在air的出现，Adobe又回到了客户端的老路上来了。Macromedia,adobe总是想把自己在桌面的优势发展到服务端，尽管推出了很多服务端产品，但是到了最后，它们又回归了客户端。在别人的博客看见对方列举了几个从Java转到Ruby有几个强有力的理由：(实际上这里他混淆了rails和ruby) 1、 Rails可以让单个开发者在开发效率方面接近自己能力的极限。我并不是一个完美架构的fans，热衷于不写一行代码，巨细无靡地分析各种架构优劣直到脑神经瘫痪。我热衷的是解决用户真正关心的问题，用技术改善他们的生活，与用户建立起良好的人际关系。 2、 Rails框架对Ajax有更好的支持，通过使用RJS模板可以极大减轻一般情况下做Ajax开发的复杂性，提高开发的效率。 3、 Rails框架对REST开发提供了最好的支持。REST在Web开发方面会越来越重要，全面支持REST就是未来所有语言Web开发框架发展的方向。 4、 JRuby 1.0对Rails的支持已经很好，而且支持直接使用Java开发的资源，所以现在是从Java转到Ruby的一个适当的时机。对于第一点：我完全同意，我一直认为面向解决方案，而不是面向编程语言，是一个开发人员应有的思路 对于第二点：

正确，rails的确简化了ajax，尽管我个人认为ajax只是一个过度的技术 对于第三点：部分认同，REST的确在Web开发方面会越来越重要，但是Rails是最早提供REST支持的框架，未必就是最好的，后来者居上的例子很多了，相信Java在这方面还是有优势的。对于第四点：Sun支持JRuby只是吸引rails开发人员将程序移植到Java平台，而不是反方向。而且JRuby,JPython等技术只是在集成上有优势，对于web开发或者不打算将应用移植到Java平台的人没什么吸引力。 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](http://www.100test.com)