

经典文萃：JavaEEwithoutAjax PDF转换可能丢失图片或格式，
建议阅读原文

https://www.100test.com/kao_ti2020/253/2021_2022__E7_BB_8F_E5_85_B8_E6_96_87_E8_c104_253840.htm 3年前，“Spring之父”Rod.Johnson写了一本在Java界引起轰动的书：《Expert One-on-One J2EE Development Without EJB》。这本书阐述了EJB作为J2EE核心技术所带来的意义与价值，但作者用了更大篇幅介绍EJB的一些缺陷与不足，并提出了Without EJB的解决方案。正是由于“J2EE Without EJB”这个激动人心的口号及这本书奠定的基础，导致了Spring Framework这个经典轻量级框架的诞生。2年前，Ajax开始进入人们的视野。时至今日，Ajax已经成为一个红得发紫的技术。但是今天，我想说一句：JavaEE without Ajax。Ajax的“原罪”Ajax为什么这样红？有人说，是因为起了个好听易记的名字（比如荷兰著名的Ajax球队，即阿贾克斯）；也有人说，是因为Google全新的Ajax应用产品给人们带来的超酷体验（比如伟大的Google Maps、GMail等）。确实如此，Ajax能够如此流行的最主要原因就是它带来了更好的用户体验，改变了人们对传统Web应用的不佳印象。然而，即使Ajax的狂热Fans也不得不承认的是，从技术层面上来说，Ajax并没有带来什么新鲜的东西。它本质上是一种新瓶装旧酒的技术，好处是通过Java Script与DHTML提供了一种异步编程模型，从而使Web应用给客户带来了更好的人机体验。正如我在去年引起大家争论的拙文《Ajax，只是一种过渡技术》中表述的：Ajax解决问题的层面较低。或者说，它解决问题的方法与手段，很难形成一种可高度抽象的框架级解决方案。并且，正是因为Ajax基于Java

Script，因此不可避免地带来了Java Script的诸多缺点，譬如：
? 跨浏览器是一场噩梦？对搜索引擎的支持不好？干掉了Back、History等按钮（尽管我并不认为Back、History是什么好东西）？开发与维护成本过高 要Java, 不要Java Script We Love Java, Not Java Script。套用毛泽东的惯用句式就是：“要Java, 不要Java Script”。相信很多读者看完这个标题也许会不以为然，但这句话却代表了许多J2EE开发人员的心声。众多Java工程师都对Java有一种近乎偏执的喜爱，他们热爱Java的简洁与优雅。但一旦让他们去进行Java Script的开发，却往往会不知所措：过度灵活的语法，无法通过编译器进行语法校验，缺乏良好的调试工具等等这些，都会让人们们对Java Script畏手畏脚，更遑论Ajax的开发。一句话，Java社区需要Ajax，需要它来提升基于JavaEE的Web应用的人机体验；但是，人们并不喜欢Ajax目前的开发模式。无疑，我们需要一种新的解决方案。谁来拯救JavaEE的Ajax？我给出的答案是JSF。目前，关于JSF的一种流行说法是“悲剧人生：Sun让JSF光着身子降临到Java Web世界”。然而，我的看法却是：作为一种革命性的服务器端组件技术，JSF犹如早晨八九点钟的太阳，前途不可限量。让事实说话，我们先来看看JSF请求/响应过程的标准生命周期：任何一个JSF“Faces Request”请求，经过Restore View、Apply Request Values、Process Validations、Update Models、Invoke Application等阶段以后，产生了一个“Render Response”返回给客户端。那么，常规JSF引擎是如何实现上述过程的呢？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