

集Java和RoR之所长的JRuby 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__E9_9B_86Java_E5_92_8C_c104_245780.htm JRuby是面向Ruby、基于Java虚拟机(JVM)的一种解释程序，它结合了Ruby语言的简易性和功能强大的JVM的执行机制，包括与Java库全面集成。Rails彻底加快及简化了Web应用的开发，不过它让人觉得不够成熟，特别是在高端企业级功能方面。另一方面，Java平台及其虚拟机、库和应用服务器的速度、稳定性和功能方面却一直在提升，现在已被公认为是开发高端服务器应用的领先平台。不过如果Java平台不与Ruby等新兴语言联系在一起，就有可能落后于流行趋势。JRuby结合了所有这些技术互为补充的优点，有望提高Ruby和Rails的知名度，同时为Java平台在运行非Java语言方面赋予新角色。

Rails：Java框架的发展方向

对Java开发人员而言，Rails就像是自然代表了诸多Java Web框架的发展趋势：减少不必要的代码、采用更多的抽象和动态机制，以及更全面的即开即用功能。约定优于配置 早期版本的Java平台企业版(Java EE)每个组件需要有大量的配置和代码。譬如说，Enterprise JavaBeans的每个bean要有多个源代码和XML配置文件。这种复杂性使得EJB成了重量级开发的代名词，最终导致EJB 3出现了180度大转变：力求普通Java对象(POJO)bean的冗余和配置最小。即使如此，重型Java EE应用程序仍需要开发人员开发代码来表示多个软件层(包括GUI、业务逻辑和持久层)上的同一业务对象。然后，尽管层与层之间存在冗余性和相似性，开发人员仍必须用配置文件把这些层粘合起来。相比之下，像Seam和Spring这些比较新的Java

Web框架使用极少的配置和代码，就可以发布业务对象。Java框架也一直在向跨Web应用程序的多个层对堆栈进行标准化和集成迈进。在早期，Java Web应用开发人员手工编写代码，从服务器小程序获得HTML输出；创建自己的模型-视图-控制器(MCV)架构，并使用SQL而不是Java数据库连接(JDBC)来访问数据库。后来，他们聚集了执行大部分通用功能的组件，譬如标记库、Struts和Hibernate。最近，Spring将大部分功能集成到了单一、自上而下的轻型堆栈。从一开始，Rails就体现了这些简洁性原则，这些原则在Rails社区中称为“不要重复自己”和“约定优于配置”(避免冗余和有意义的默认值是软件工程领域中的两条古老原则)。该框架可以根据简明的约定，猜出不同层的连接关系。Rails甚至可以显式、动态添加属性从而反射数据库列: last_name列会自动使last_name属性出现。在约定不能满足要求的特殊情况下，仍可以使用纯Ruby代码或者类似Ruby的轻型YAML格式来添加配置，这两种格式都删去了XML的冗余方括号和结束标记。动态和反射机制

Java框架也一直在向反射和元编程机制使用更广泛而迈进。譬如说，Spring使用反射机制，利用依赖注入把各部分连接起来；标准的Java EE服务器系列则通常采用静态方法。Hibernate这种流行的对象关系映射框架利用动态元编程进行映射，在运行时更新字节码，而不像早期的数据访问框架需要在开发时生成大量的源代码或者字节码。Hibernate的开发人员之前只好采用一些高级技术来实现这项功能。而在Ruby中，元编程却是这种语言的一个有机部分，结果Rails在运行时不但能动态生成映射，还能生成访问及显示底层数据库所需的业务层类定义，从而尽量减少了这种需要。支持

开发过程 上世纪90年代末前后，Java开发人员大量使用JUnit框架的测试方法，但为服务器端应用程序编写测试用例总是有难度的。如今Spring在生成Web应用程序的同时还能生成测试。Rails具有同样功能，充分利用了动态机制和元编程技术来支持多种测试:单元测试、功能测试以及集成测试。为什么在JVM上运行Rails 如果你现在正在使用Spring和Hibernate等Java框架，那就不用不着改变。但如果可以灵活地为下一个项目选择新的开发方法，不妨考虑Rails。遗憾的是，改用一种新语言一般被认为是危险举措，管理人员对风险有顾虑。JRuby能够让Rails更容易被管理人员所接受。在JVM上，Rails成了一种Java框架。譬如说，“Java”Web框架中的非Java语言实际上同样用于JavaServer Pages中。就像通常用C/C++编写的操作系统为使用比较抽象的语言编写的应用程序提供了基础架构一样，Java平台也为Ruby等动态语言扮演“系统软件”的角色，提供了基础架构层面的支持。如今可以通过Java访问众多的功能。JDBC和Java消息服务(JMS)等API是同类中最佳的，而许多不可替代的内部或者独立软件开发商的企业信息系统可以通过Java API来访问。Rails应用程序通过使用JRuby，可以像调用Java代码那样调用现有的Java库。有了JRuby，Rails应用程序可与Java Web应用程序在现有的Java EE应用服务器上一起运行。这种应用服务器拥有强大的技术基础架构。在人员和培训方面，通常不缺乏教育计划以及有经验的开发和支持人员。另外只要运行在JVM上，这种应用服务器就能够获得最近十年在JVM方面投入的许多优化项目所带来的好处。JVM拥有比Ruby复杂得多的安全模型，为JRuby on Rails提供了处理Web应用程序的常见安全难题的工

具，包括控制从各个来源获得的Ruby脚本。它还包含了内部支持国际化的功能。向动态语言迈进，这是在抽象机制方面向更高级语言发展的一个方面。在JVM的管理环境下而不是在操作系统层面上运行Ruby解释程序又是向更高抽象层迈出的一步。使用Rails作为Web应用程序层的标准实现机制也是如此，只把针对特定应用程序的功能留给了开发人员。链接：[JRuby面临的难题](#) 正如版本号所示，JRuby 0.9.2还没有准备好运行生产应用程序。一些错误有待解决；另外，目前JRuby的速度不如MRI。与Rails一起使用Java应用服务器需要非标准的适配器服务器小程序，而构建war文件需要特殊的Ant脚本，这两者还不是JRuby发行版的标准部分。 Rails在处理遗留组件方面的功能特别弱。虽然Rails为解决大多数常见问题提供了很好的支持，但缺少支持替代方案的灵活性。譬如说，活动记录假定每个表都有一个名为id的单一主键列。虽然可以用键列代替另一个名称，要是不使用特殊插件，就无法定义多列键。相比之下，Hibernate等Java框架虽然在简单(且常见)的情况下开发速度比较慢，但处理极端状况和遗留代码的效果比Rails好得多。最终，采用Rails面临的主要难题在于，是否渴望现有语言和框架方面统一标准。 Rails适用于这种渴望比较强烈的新项目和新组织。 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