

Javascript:关注JSVM-java基础 PDF转换可能丢失图片或格式，  
建议阅读原文

[https://www.100test.com/kao\\_ti2020/471/2021\\_2022\\_Javascript\\_c67\\_471291.htm](https://www.100test.com/kao_ti2020/471/2021_2022_Javascript_c67_471291.htm) 介绍 我们在 JSVM1.x 中尝试了一种较传统方式不同的 JavaScript（基于web中的JS）开发模式。一年多的时间中，得到了大量积极有效的反馈和建议，在这些基础上今天我们正式推出 JSVM2。 JavaScript 一直以来扮演着Web端“小脚本”的角色，开发者用它来在web上制作一些动态特效，以及表单提交前对数据的预校验等等，随着客户端处理能力日益提升，DHTML在表示层的动态特征和丰富表现力，以及XmlHttp,XmlDom等HTTP工具的出现,另外一种应用模式开始引起人们的关注：基于 JavaScript DHTML XML

XmlHttp/XmlDom 的 RIA (Rich Internet Application)，更确切地说，应称为：RWA (Rich Web Application)。这类应用给用户带来的良好体验远胜于传统 Web 中依赖 Form 提交数据的方式，并且较大程度减小了服务器的负载和网络流量，提高了系统综合性能。后来一个老外提出了 Ajax 这个名词，关于这个概念我不再冗余复述。如今 Ajax Framework 像雨后春笋一般涌现很多，其中不乏优秀之作。但在此并不多作介绍。因为 JSVM 和他们定位不同，JSVM不仅仅是一个 Ajax 框架，也并非仅仅为了实现 Ajax 而产生。它作为一个更底层更抽象的 JavaScript 框架，为我们实现类似 Ajax 应用提供了一个基础平台。我曾经拿jsvm1到51js上与许多js爱好者共同讨论，其中有两种比较典型的不同看法是：JavaScript 需要框架吗，是不是把简单的事情复杂化了？短小精干才是脚本之道。非要借鉴 JAVA 的架构吗？是否有更好的解决方案。对于第一种意见，

我认为某种条件下还是有道理的。例如：我仅仅是想打印一句 Hello, world! 直接 alert() 就可以了，何必要弄一个 Framework 出来，岂非劳民伤财，小题大做。确实，对于一些简单的 web 应用，js 在整个体系中充当的角色很小，我觉得是可以不用框架的。但对于一个 RIA 模式的 Web 应用，尤其是企业级的应用，系统复杂程度和 JS 代码规模相当庞大的情况下，没有框架带来的后果是可以预见的。对于第二种观点：赞成框架的必要性，但对 jsvm 一些“模仿”java 的做法抱有怀疑态度。我不能说这种看法不对，也许确实有更好 API 架构。但是，本人认为这个不是关键所在，jsvm 提供了对多种语法解析器的支持，有兴趣的人完全可以根据自己喜好在 jsvm 下开发出另外一套 API。类 C#, 类 VB 的语法都可以实现。简单的说，JSVM 根本目的是为了规范浏览器端 JavaScript 脚本开发，更好的实现代码复用。传统面向过程型的 JavaScript 开发方式已经明显不能满足大规模软件开发的需要。大部分迹象表明？随着 RIA 的发展，客户端都将承担更多的逻辑处理工作，企业级应用的代码规模和复杂度都是相当大的。有效的代码组织结构，合理的模块粒度，低耦合高内聚的设计原则等等如何引入到 JavaScript 的开发上来？相信当你了解 JSVM2 之后，会有一些答案。作为一个框架，JSVM 要解决的问题不是提供某一种应用的解决方案，而是针对各种场景提供广泛的支持。另外一个需要说明的是：jsvm2 并不是 jsvm1 的一个升级版，几乎是在 jsvm1 上完全重构。且不提供对原 jsvm1.x 中的类的支持。（当然，原 jsvm1.x 下的类需要经过做一些小改动即能在 jsvm2 下正常工作）较 jsvm1.x，jsvm2 有了几处明显的变化性能上有了明显的提升。每个类文件拥有独立的语法域。几

种常见的web运行模式在Framework中将直接被支持。编译引擎支持多语法解析器，并缺省提供了一个名为"jsvm2"的语法解析器。从语法层面提供了一些OO关键字的支持，使jsvm下的面向对象的javascript开发更为直观和方便。具有更灵活的扩展性。二次开发者可以很方便的扩展其功能。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)