基于Java的web开发框架的历史、现在与将来

浙江工业大学,计算机学院,软工XX

摘要：Java Web应用开发技术可简述为Java Web应用程序供用户通过浏览器发送请求，在web服务器上运行程序，产生web页面，并将页面传递给客户机器上的浏览器，将得到的web页面呈现给用户。Java Web主要用来开发一些大型的网站,由于Java Web有他不可替代的优势,所以现在用Java做网站是比较流行的。Java Web开发框架的应用给人们的工作、生活以及学习带来了极大的便利，前景优势很大。

关键词：Java Web技术；框架；便利性

The History, Present and Future of the java web Development Framework

ZhuKexin

Zhejiang University of Technology, Computer Institute ,Software Engineering Class1804

**Abstract:** Java web application development technology can be briefly described as java web application for users to send requests through browsers, run programs on web servers, generate web pages, and pass pages to browsers on customer machines to present the resulting web pages to users. java web is mainly used to develop some large websites, because java web have his irreplaceable advantages, so it is more popular to use java to do websites now. Application of java web development framework brings great convenience to people's work, life and study, and the prospect advantage is great.

# 1 Java Web开发技术的发展阶段

## servlet技术

Servlet是Java技术中最早的Web解决方案，Servlet与普通Java类的编写非常类似。使用Servlet技术，页面中的所有信息需要通过输出语句来生成。下面的代码（部分代码）生成的页面的内容就是”HelloWorld!”[4]。

**public** **void** doGet(HttpServletRequest request,HttpServletResponse)

**throws** IOException,ServletException{

response.setContentType("text/html;charset=gb2312");

PrintWriter out = response.getWriter();

out.println("<html>");

out.println("<head><title>Hello World！</title></head>");

out.println("<body>");

out.println("<p>Hello World！</p>");

out.println("</body></html>");

}

## JSP技术

JSP技术的目标是快速开发网站，在网页中嵌入Java代码完成网站。例如下面的代码就是1个简单的JSP例子：

<html>

<head><title>测试</title></head>

<body>

第一阶段<% String str = “test” ; out.println(str); %>

</body>

</html>

如果希望进行一些处理，直接在界面中使用<% %>脚本标识嵌入Java代码。如果要开发具有大量网页内容的网站效率比较高，可以先使用网页编辑工具编写网页，然后在网页中嵌入处理代码即可。但是缺点是：网站的输入输出、处理、控制全部夹杂在一起，维护不方便。当网站中需要进行大量的处理代码的时候，JSP文件将很难维护，并且代码也不容易共享。

因为JSP在编写网页方面具有优势，而编写处理代码存在很多问题，所以产生了模型1，把JSP中的处理代码使用JavaBean来实现[4]。

## JavaWeb模型阶段

1. JSP+JavaBean

JavaBean是使用Java语言编写的组件，简单的理解，可以认为JavaBean是比较特殊的Java类。JavaBean会把网站中处理过程进行封装，这样的结果就是在JSP文件中仅仅是输入输出功能和控制功能，处理过程放在JavaBean中，然后在JSP中调用JavaBean的功能，JSP提供了若干标签支持调用。

例：

JavaBean文件：

**package** test;

**public** **class** PrintMessage

{

**public** String print()

{

**return** “使用JavaBean返回信息”;

}

}

页面文件：

<%@ page import=*”test.\*”* %>

<jsp:useBean id=*”printMessage”* class=*”test.PrintMessage”* scope=*”request”*/>

<html>

<head><title>测试</title></head>

<body>

第一阶段<% out.print(printMessage.print()); %>

</body>

</html>

这种模型的优点是：业务处理由JavaBean完成，减少了页面中的代码。缺点是：页面中仍然有用于控制的代码[4]。

1. JSP+Javabean+Servelet

在这种模型中，把JSP中的控制功能提取出来使用Servlet实现。这样，JavaBean强调的是功能，Servlet主要用于控制。从完成的功能来说，JSP和Servlet是相同的，但是JSP主要以标签（HTML标签和JSP标签）为主，Servlet是Java代码。JavaBean和Servlet都是Java代码，但是Servlet能够接收请求，用户能够直接访问，JavaBean不能接收请求，需要由JSP或者Servlet调用。

这种模型的特点：JSP完成输入和输出，主要使用HTML标签和JSP标签。Servlet主要完成控制，接收请求，调用业务处理（JavaBean），根据调用的结果找到一个JSP页面对用户进行响应。JavaBean完成所有的处理过程。

这种方式实际上是MVC模式的采用。MVC模式由3部分组成：M表示模型（模拟现实事物），主要用于处理过程；V表示视图，实际上就是输入和输出功能；C表示控制器，主要控制程序的流程[4]。

## 框架阶段

随着Java Web应用技术的发展，出现了大量的框架包括Web层框架和持久层框架。

Web层框架通常可以提供一些Web应用的通用功能，用户在编写应用的时候只需要关注自己比较特殊的东西即可。例如，Servlet作为控制器完成的功能包括：获取用户的输入信息；对信息进行验证、转换；使用这些信息调用业务逻辑（JavaBean的方法）；然后选择界面对用户响应。如果把获取的信息设置成变量、要进行的验证和转换进行配置、要调用的方法进行配置、要响应的界面进行配置，这样Servlet就可以通用了。在本书的第二部分对流行的Web层框架Struts2和JSF进行了介绍，并自己实现了1个简单的框架。

持久层框架，主要是解决数据库访问问题，同样是把数据库访问中通用的内容提取出来，在编写应用的时候只编写特殊的东西。例如，要把用户信息和图书信息添加到数据库中，采用JDBC技术，编写的两段代码的区别是表名不同，列与属性的对应关系不同。持久层框架就可以完成这个添加工程，用户只需要提供表名并且提供列与属性的对应关系。书中第二部分持久性部分将对持久层框架进行介绍[2]。

## 可视化开发

随着集成开发环境对于Web应用支持的增强，可视化Web应用开发（JSF技术）应该是未来的发展趋势。这种方式的开发就像以前在C++Builder或者VB中开发应用一样方便。

# **2 JavaWeb框架介绍**

Web系统总体架构被分为2大部分：

* 客户端：基于浏览器提供信息展现、用户交互等功能。所采用的技术主要有：HTML/HTML5、Javascript、CSS。另外，Flush由于其广泛的浏览器集成度，通常也可归纳为一种WEB技术，但Flush不在本文讨论范围。
* 服务端：实现业务逻辑处理。通常按三层架构模式划分为展现层、业务逻辑层和数据集成层。服务端的平台选择相比客户端来讲更加广泛，有PHP、Java、.Net、Ruby、Python等。每种平台下都有非常优秀的WEB框架可供选择。

尽管客户端的WEB技术平台主要还是基于HTML＋Javascript＋CSS，但是基于其上的类库、框架、开发模式、衍生技术等非常繁杂且变化多端，要在实际项目中做出最好的选择绝非易事。有鉴于此，本文不打算深入讨论客户端WEB技术。

服务端虽然平台选择非常多样化，但总体架构基本一致。本文主要讨论Java平台。在所有平台中，Java平台下的WEB框架是最多的，其它语言平台下新的框架总能在Java中找到对应的实现。这也是Java平台生命力强大的体现之一。

一些主流web框架介绍如下[3]：

* Struts：这是Java社区最老牌、知名度最高、使用也最广泛的WEB框架。Struts的特点是简单易用、文档丰富，通常与spring-hibernate/ibatis组合使用（SSH／SSI）；
* Spring MVC：Spring当年凭借一句"J2EE Withou EJB"的怒吼拉开了开源社区全面对抗学院派（JCP）的序幕，这一战让Spring一举成名。凭借Spring的东风，其mvc框架也得到社区广泛关注。基于Spring的IOC、AOP等技术，其框架设计简洁优雅、扩展性非常强；
* Play! Framework：这是Ruby社区非常火爆的Ruby On Rails框架在Java平台的翻版。当年Ruby On Rails的约定优于继承、极简的ORM框架所带来的快速开发能力与当时Java社区经典的SSH中繁杂的XML配置、越来越臃肿复杂的ORM所导致的开发效率的低下形成了鲜明对比。Play!就是这种理念PK下的产物；
* EOS：国产的基于构件理念的Java开发平台，其图形化的业务逻辑编排能力让许多人眼前一亮，对SOA不遗余力的支持也体现了厂商的态度。EOS的页面流概念应该算是WEB框架领域的一个创举，此前或此后还没有哪个WEB框架考虑过多个请求之间的关联性问题。除此之外，EOS的WEB框架也算中规中距；
* JSF：2001学院派发起JSR，至2004年推出规范1.0、2006年推出1.2版参考实现、2009年发布2.0规范、2010以后JSF2得到了以JBoss为首的广泛支持。然而时至今日，JSF仍然不温不火，其成就远没有Struts或Spring MVC来得高。不过个人认为，相比前面所说的所有框架来讲，JSF的设计理念是非常先进的。JSF是Java世界中非常少见的以组件为中心的WEB框架！补充一点，金蝶2007年推出的OperaMasks当年高调宣称其基于JSF的框架是世界一流的，但时至今日其官方网站几乎停止更新，着实让人不甚唏嘘；
* GWT：Google出品的Java WEB框架，倡导使用传统桌面应用开发方式来开发WEB。开发人员不需要懂WEB技术，只需要熟悉Java和面向对象理论，就可以使用类似Swing或RCP的方式开发WEB应用。这一点与eclipse的RAP非常相似。GWT适用于从传统桌面开发转向WEB开发的人群，对于真正了解WEB技术的开发人员来讲，其开发模式很难让人接受；类似这样的框架还有apache wicket等[2]。

# **3 JavaWeb框架前景浅析**

尽管Java Web框架各类繁多，不同的框架有不同的特点，但不同的框架之间还是有许多共性的。例如，按照页面和处理逻辑的关系我们可以将Java WEB框架划分为：

传统MVC模式的WEB框架：如Struts、Spring、Play!等。这一类框架的特点是页面和处理逻辑按照传统MVC模式进行组织，页面通常使用JSP或某种模板语言（如Freemarker、Velocity等）来实现，服务端处理逻辑通常采用Action或Command模式[5]；

以组件为中心的WEB框架：如JSF。传统ＭVC框架的请求处理流程通常是“请求-处理-页面”的循环，而在JSF中请求的处理是基于事件的，这种开发模式类似于桌面程序的开发模式，只是页面还是使用JSP加JSF标签组件而已；

模拟桌面开发方式的WEB框架：如GWT、Wicket等。这类框架纯粹使用桌面方式进行ＷＥＢ开发，开发人员基本不需要知道HTTP及HTML/JS/CSS等知识。这类框架相比JSF更加激进，不仅请求处理模式基于事件，连页面展现也是按照传统桌面方式去开发；

上述三类框架中，第一类是最靠近WEB的开发方式，第三类是最靠近桌面的开发方式，第二类界于两者之间。桌面开发方式的好处是组件化能力非常强，借助发展多年的桌面控件设计经验，可以很容易地设计出复用度非常高的组件。相对的，第一类开发模式下由于界面变化非常大，在展现层就很难做出组件化的设计（这其实也体现了WEB应用展现及交互方式变化多端的内在本质）。

另一方面，第一类框架非常强调客户端与服务端的分离，而第三类框架则试图弱化客户端与服务端的界限，其理想模式是客户端事件直接传递到服务端，中间没有任何转换（这就是桌面程序的模式了）。

考虑现实世界的复杂性以及WEB千变万化的特点，第一类框架只是做好自己的份内事，将客户端的处理交由专门的客户端框架去实现（这样才能充分利用当前欣欣向荣的客户端WEB技术）；而第三类框架则试图通吃客户端与服务端，完全无视客户端的特点，这将导致第三类框架只能局限于某些特定领域的应用范围之内，而且随着客户端WEB技术的发展，其应用范围必定越来越狭窄。至于第二类框架，界于前两类框架之间，地位非常尴尬，尽管其有官方正统的血统背景，但其违背WEB大的发展趋势，其前景不被看好。[3]

Phone手机用户，他还可以直接吸引数十亿Windows PC用户。

# 4 结束语

Java Web主要用来开发一些大型的网站,由于Java Web有他不可替代的优势,所以现在用Java做网站是比较流行的。Java Web开发框架的应用给人们的工作、生活以及学习带来了极大的便利，前景优势很大。

Java web的优势在于：

1. 安全性：有java做强有力的后盾支撑，安全性高；
2. 广泛性：不但有成熟的设计模式，而且还有成熟的框架，可以用很多表达式以及标签来展示我们需要的内容；
3. 稳定性：安全性高，也是它具有稳定性的原因之一，另外，因为java Web被用来做很多大型的商业项目，所以说，稳定性比较高，另外还有很多高手不断的改进。

参考文献

[1]张学敏. 浅议MVC设计模式在JavaWeb中的作用.

[2][周恕义](http://www.cqvip.com/main/search.aspx?w=%e5%91%a8%e6%81%95%e4%b9%89),[王欢](http://www.cqvip.com/main/search.aspx?w=%e7%8e%8b%e6%ac%a2).[基于Web框架的新一代网络教学平台研究](http://www.cqvip.com/QK/87066X/200808/27869243.html" \t "http://www.cqvip.com/main/_blank).

[3]2019年Java Web最流行的开发框架总结--CSDN博客

[4]java的江湖——对基于java的web应用开发之整体认识--CSDN博客

[5]Play! 一个颠覆Java EE规范的Web开发框架--《程序员》 2009年 第12期