# Spring Web MVC——快速开源的J2EE应用开发框架：一个简单的实例

Praveen Gupta

Research Scholar, Singhania University

Pacheri Bari, Rajasthan, India

Prof. M.C. Govil

Govt. Mahila Engineering College

Ajmer, Rajasthan, India

摘要——Web应用程序开发在现在高速发展。开发出精准的、经济的和效率的程序是需要时间的。我们有兴趣提高生产力和降低复杂性。这在改变程序员开发Java 2平台, J2EE Web应用程序方面是一个潜在的主题。我们的重点是不用EJB如何创建兼容j2ee的软件。最好的选择之一便是Spring框架,它比EJB提供更轻量级的服务。这一转变背后的驱动力是需要开发和实现强大生产力并降低复杂性的Web应用程序。在本文中,我们简要描述Spring的底层架构和一个简单的案例研究。

关键词：MVC, Spring, XML

## I．介绍

现在的网络是非常复杂的。当公司和组织的要求增加，网络编程的复杂性和性能问题就越来越凸显。复杂的不同类型的通讯设备越来越多。业务是用网络和许多的通信设备搭建的应用程序。因此，我们不得不关心架构问题来应对在互联网上增加的数据负载。让我们来讨论使用spring web mvc框架是如何快速开发应用程序的以及模型视图的体系结构。

Spring框架提供了丰富的功能,让我们在简短的讨论这些特性。

1. 控制反转：控制反转(IOC)是用于为一个应用程序提供服务或组件的技术。IOC是一个在相比传统的交互模式下系统的控制流是倒转的软件设计模式和配置相关的编程技术。IOC不是应用程序的调用框架，而是通过应用程序调用组件的一个框架。IOC可以被解释为“请求资源的注入或者依赖在运行期注入相关资源”其也被称作依赖注入。org.springframework.beans.factory.BeanFactory是Spring IoC容器负责包含和管理bean的实际的类。BeanFactory接口在Spring中是Spring IoC容器的核心接口。一个bean是一个通过Spring IoC容器管理的简单的实例化对象。这些bean和它们之间的关联反映在通过容器的数据配置上。
2. 构造函数注入：我们可以使用Java类的构造函数来加载bean的值。一个Java类是  
   通过单一的构造函数定义的。 Details.xml文件是通过构造函数提供值。 现在Java使用另一种bean工厂的方法加载XML文件。java在构造函数中使用XML文件来加载其中的值。这是用来将值传递给构造函数。
3. 属性注入：我们为每个bean定义了getter和setter方法。我们也可以使用setter方法设置bean中的值。Setter方法会覆盖从bean中加载的值。
4. 接口注入：在Spring中我们也可以定义接口类。为了实现这一点，我们将接口导入到java程序中。现在我们可以在Spring和XML中用接口来定义方法。
5. 继承：一个java类可以获得另一个类的属性,就像一个java程序。这里有三个子类型。1.抽象类:bean声明为抽象的在Spring中不能被继承。2.父子类:我们可以像父母和孩子一样定义层次结构。3.父-子-子子关系:我们可以为更多的类定义层次结构。
6. Autowiring：自动装配被用于xml文件和java文件的属性名、值对映射。有四种集成类型。通过名字,通过类型,通过构造函数和自动检测。如果没有定义那么默认为通过名字来装配的。
7. Bean的生存周期：在Spring里，所有bean定义有四个原型值的范围,会话、请求,单例,全局会话。这是被用来控制访问的bean的。
8. Bean的引用：一个xml文件中的bean可以从其他bean中被指定值。这被用于从一个bean读取值,并分配到另一个bean中。

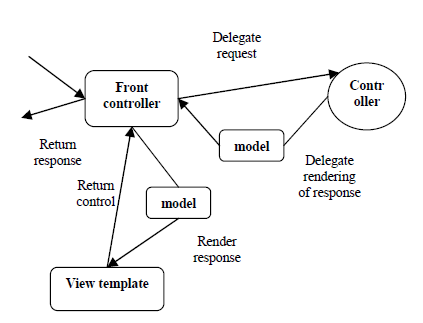
## II．Spring的主要组件

Spring也遵循MVC的主体架构并包含三个核心组件。1. Controller:处理导航逻辑和为业务逻辑进行服务层交互2. Model:控制器和视图之间的契约。包含的数据通过控制器发送给视图进行渲染。3. View:呈现请求从模型中获取的数据响应。Spring MVC的核心组件如下。

1. DispatcherServlet:这是Spring的前端控制器实现。Web.xml接收请求,并转移给DispatchServlet。这是进行请求交互第一个控制器。它是用Servlet实现的。它完整的控制应用程序的导航流程。
2. Controller: 这是用户创建用来处理请求的组件。它封装了导航逻辑。Controller是服务对象的服务员。
3. View: 视图负责渲染输出。不同的视图可以基于不同设备选择不同的输出类型。
4. ModelAndView: ModelAndView是spring框架的核心部分。它实现应用程序的业务逻辑。它被Controller创建。它联合视图请求。它存储业务逻辑和模型数据。一个Controller调用它然后它将被执行。执行它将返回数据和视图的名称。
5. ViewResolver:如何显示输出取决于从ModelAndView中收到的结果。它被用来映射逻辑视图名称到实际视图的实现。这部分实现的是如何显示输出它。
6. HandlerMapping:策略接口被DispatcherServlet用来映射从传入请求到个体控制器。它标识请求,并调用相应的处理程序来提供服务。处理程序将调用控制器。

## III．Spring的架构

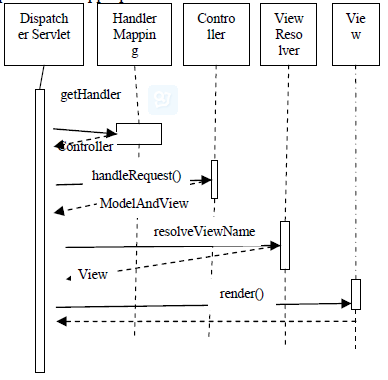
Spring框架提供了一个功能齐全的MVC模块来构建Web应用程序。spring的MVC架构可配置多个视图技术，支持JSP，Velocity，Tiles， iText等。Spring MVC将控制器,模型对象, dispatcher Servlet和处理器对象等多个角色分离。将对象和控制器清楚的分离开来，使得它们更容易定制。图1显示了执行流视图。



图一：执行流视图

下面的图2显示了Spring模型的序列图。其中dispatcher Servlet是应用程序的入口。当dispatcher Servlet得到服务请求后调用处理器。所有处理器都被映射到Servlet。处理器开始执行,调用相应的控制器并将请求参数传递给它。然后控制器执行,它包含业务逻辑和一个与控制器相关联的模型视图。执行后它将返回ModelAndView到Dispatch Servlet。这个ModelAndView包含数据和视图名。

　Dispatcher Servlet从控制器处得到包含数据和视图名的ModelAndView。然后Servlet将调用视图解析器。视图解析器通过数据提交识别视图的名称。最后它将数据以合适的格式呈现给用户。



图二：Spring框架中的应用程序序列图

## IV．Spring和XML

Xml在spring框架中被广泛使用。它简化了开发流程并且节约了时间。Xml被用来存储应用执行过程中需要的数据。Web.xml是应用的入口。它会告诉你浏览的路径。它装载了应用上下文类并且告知DispatcherServlet xml配置文件的位置。

**web.xml:**

**-** <?xml version="1.0" encoding="UTF-8"?>  
<web-app version="1.0" xmlns="http//java.sun.com/xml/ns/javaee"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:schemalLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app-2\_5.xsd">  
<listener><listenerClass>org.springframework.web.context.ContextLoaderListener</listener-Class></listener>  
<servlet>  
<servlet-name>dispatcher</servlet-name><servletclass>org.springframework.web.servlet.DispatcherServlet</servlet-class>  
<load-on-startup>1</load-on-startup>  
</servlet>  
<servlet-mapping>  
<servlet-name>dispatcher</servlet-name>  
<url-pattern>/send/\*</url-pattern>  
</servlet-mapping>  
<welcome-file-list>  
<welcome-file>index.jsp</welcome-file>  
</welcome-file-list>  
</web-app>

上面的web.xml会与服务器接收到的请求互动。这个文件定义了作为dispatcher的servlet的名称，应用上下文类和作为主页的index.jsp。\*.\*这个URL模式能够匹配所有入站请求。

**ApplicationContext.xml:** ApplicationContext构建在BeanFactory的基础之上。它能方便的与spring AOP 特性，消息资源处理，事务传播的集成。BeanFactory提供框架的配置和基础功能。ApplicationContext给应用增加了一些增强的特性。在J2EE环境中，ApplicationContext一定会被使用。

<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:p="http://www.springframework.org/schema/p"  
xmlns:aop="http://www.springframework.org/schema/aop"  
xmlns:tx="http://www.springframework.org/schema/tx"  
xsi:schemaLocation="http://www.springframework.org/schema/beans  
http://www.springframework.org/schema/beans/spring-beans-2.5.xsd  
http://www.springframework.org/schema/aop  
http://www.springframework.org/schema/aop/spring-beans-2.5.xsd  
http://www.springframework.org/schema/tx  
http://www.springframework.org/schema/tx/spring-tx-2.5.xsd">  
<bean id="superClass" class="packagename.SuperClass" />  
<bean id ="subClass" class=" packagename.SubClass">  
</bean>  
<property name="superClass" ref="superClass"/>  
</beans>

上面的文件加载了SuperClass.java 和SubClass.java的对象，同时也定义了引用的类。

**Dispatcher-servlet.xml** Spring web mvc框架是请求驱动的web MVC框架，围绕一个用于分发请求到控制器的servlet进行设计，并且提供很多额外功能。DispatcherServlet和spring的ApplicationContext完全融合以至于可以利用spring的功能。分发器是应用的中央控制单元，被用来定义视图解析器，bean，处理器和它们的映射。  
<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns=<http://www.springframework.org/schema/beans>

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:p="http://www.springframework.org/schema/p"  
xmlns:aop="http://www.springframework.org/schema/aop"  
xmlns:tx="http://www.springframework.org/schema/tx"  
xsi:schemaLocation="http://www.springframework.org/schema/beans  
http://www.springframework.org/schema/beans/spring-beans-2.5.xsd  
http://www.springframework.org/schema/aop  
http://www.springframework.org/schema/aop/spring-beans-2.5.xsd  
http://www.springframework.org/schema/tx  
http://www.springframework.org/schema/tx/spring-tx-2.5.xsd">  
<bean id="viewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver">  
<property name="prefix">  
<value>/WEB-INF/views/</value></property>  
<property name="suffix"><value>.jsp</value></property>  
</bean>  
<bean id="urlMapping" class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">  
<property name="mappings">  
<props><prop key ="/\*">dispatchController</prop>  
</props>  
</property>  
</bean>  
<bean id="dispatchController" class="packagename.DispatchController"></bean>  
</beans>

## V．Spring如何工作

快速应用程序开发是行业长久以来的需求。有很多开发的辅助工具，但同时技术和需求也在增长，所以我们需要新的工具和架构去掌控日益增长的应用。工具因架构而异。Spring采用MVC架构，可以支持大型应用。使用spring这种技术，可以方便的在原应用的基础上进行二次开发而不改动已有的代码。它使用xml文件向应用中添加映射，请求，java bean等等事物。

## VI．架构优点

让我们看一下spring web mvc 框架为项目架构带来的好处

1. Spring可以有效的组织中间件，然而EJB不可以。配置管理服务可以在架构的任何层或者任何运行环境中使用。
2. Spring web MVC框架是采用MVC设计模式，为快速开发web应用程序所设计的鲁棒，灵活且具有良好设计的框架。
3. Spring消除了单例对象的扩散。这是一个降低可测试性和面对对象设计的主要问题。
4. 清晰的角色区分：Spring MVC很好的区分开了框架中各个组件的角色。所有的组件比如控制器，命令对象都有不同的角色。
5. 可适配的控制器：如果你的应用不需要一个HTML表单，你可以写一个简单的spring控制器，它没有表单控制器需要的那些额外的组件。Spring提供很多种控制器，每一个都有自己特定的目的。
6. Spring通过在应用中使用一致的配置，从而减少了大量自定义属性文件格式的需求。
7. Spring使用接口编程而非使用类，以提供一个好的编程实践。
8. 使用spring编写出来的应用只依赖于少量的spring api。Spring应用中的大多数业务对象不对spring本身产生依赖。
9. 使用spring编写出来的应用很容易进行单元测试。
10. spring可以使用EJB作为一个实现来代替，而并非应用架构的决定因素。（这个翻译，我真是不知道该怎么理解）
11. 你可以选择使用POJO或者本地EJB对象来实现业务接口，而不需要改变调用的代码。
12. Spring可以在很多应用中代替EJB的使用。在没有EJB容器的情况下，可以使用Spring的AOP来实现申明式的事务管理。
13. Spring提供为数据存取的一致性框架，不论底层使用的是JDBC还是ORM框架，如hibernate。Spring在很多领域比如JDBC，JMS，JavaMail，JNDI提供了一支且简单的编程模型，同时提供了很多API，这些让它成为一个理想的框架。
14. Spring采用POJO来构建应用，这为开发者屏蔽了复杂性。
15. Spring可以让你无需关注JDBC连接的泄漏问题，我们只需要书写一些必要的SQL语句，它同时也能处理从数据库返回的错误信息。

## VII．总结

Spring web mvc框架可以在RAD环境中为应用程序提供运行环境，也可以保证程序的一致性，性能和可靠性。我推荐开发者在大型web程序环境也采用该框架，因为它是开源的。

参考文献：

[1] Shu-qiang Huang, Huan-ming Zhang, ” Research on Improved MVC Design Pattern Based on Struts and XSL” , in Information Science

and Engineering ISISE 08 International Symposium on, 2008, vol. 1 PP. 451 – 455

[2] Juanjuan Yan; Bo Chen; Xiu-e Gao, "Le Wang; Research of Structure Integration Based on Struts and Hibernate" , in 2009 WRI World

Congress on Computer Science and Information Engineering,2009, vol. 7,PP. 530-534

[3] Wojciechowski, J.; Sakowicz, B.; Dura, K.; Napieralski, A.,"MVC model, struts framework and file upload issues in web applications

based on J2EE platform", in Proceedings of the International Conference on Modern Problems of Radio Engineering, Telecommunications

and Computer Science 2004, 2004, ,PP 342-345

[4] Erxiang Chen; Minghui Liu, "Research and Design on Library Management System Based on Struts and Hibernate Framework", in WASE

International Conference on Information Engineering ICIE 09, 2009, Vol. 2,PP. 310-313

[5] Yonglei Tao; "Component- vs. application-level MVC architecture", in Frontiers in Education 2002 FIE 2002. 32nd Annual,2002, Vol 1,PP.

T2G-7 - T2G-10

[6] Meiyu Fang, "Design and Implement of a Web Examination System Using Struts and EJB" ,Seventh International Conference on in Webbased

Learning 2008, ,, 2008,pp. 25-28

[7] Wang Ning; Li Liming; Wang Yanzhang; Wang Yi-bing; Wang Jing,"Research on the Web Information System Development Platform

Based on MVC Design Pattern",in IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 2008 ,

Vol 3,pp. 203-206

作者简介

**Praveen Gupta**在印度的Nagpur大学获得电子工程学学士学位,在Punjabi大学获得信息科学硕士学位并在Singhania大学读博士。他在孟买的一个软件公司做技术主管。是一位研究学者，计算机电子协会的会员和电气工程师。主要研究方向包括： MVC模式, Java框架和设计模式。

**Prof. M.C. Govil** 在IIT获得了博士学位.是大学教授和女性工程学院的负责人。他在教育和研究领域有超过20年的经验.是很多协会的会员并发表了大量的论文.主要研究领域是： 通讯, 网络技术，软件工程等。