

本科毕业论文（设计）

英文文献翻译

|  |  |
| --- | --- |
| **文献英文题目** | **Spring Web MVC Framework for rapid open source J2EE application development: a case study** |
| **学 生 姓 名** | **袁曦焜** |
| **学号** | **2015081147** |
| **专业** | **软件工程** |
| **年级班级** | **2015级4班** |
| **指导教师** | **杜晓宇（讲师）** |
| **所在学院** | **软件工程学院** |
| **提交日期** | **2019年5月22日** |

2018 年 12 月

成都信息工程大学 软件工程学院

关于Spring Web MVC框架用于快速开发J2EE程序开发的案例研究

Praveen Gupta

Research Scholar, Singhania University

Pacheri Bari, Rajasthan, India

Prof. M.C. Govil

Govt. Mahila Engineering College

Ajmer, Rajasthan, India

摘要： 如今，Web应用程序的开发竞争激烈，需要时间准确，经济，高效地开发应用程序。 我们有兴趣提高生产力并降低复杂性。 这是改变程序员开发Java 2平台企业版（J2EE）Web应用程序的方式的一个基本主题。 我们的重点是如何在不使用Enterprise Java Beans（EJB）的情况下创建符合J2EE的软件。 最好的替代方案之一是Spring框架，它提供的服务更少，但它比EJB更少侵入。 这种转变背后的驱动力是在Web应用程序软件开发和实现领域需要更高的生产力和更低的复杂性。 在本文中，我们简要介绍了Spring底层架构，并使用Spring Web MVC Framework进行了案例研究。

关键词：MVC;Spring; XML

# 介绍

如今，Web是一个非常复杂的问题。 由于公司和组织的愿望在增加，因此Web编程的复杂性和性能很重要。 与不同类型的通信设备的复杂性正在增加。 该企业要求使用网络和许多通信设备的应用程序。 因此，随着互联网上数据负载的增加，我们必须处理架构问题。 让我们讨论一下如何使用spring web mvc框架快速应用程序开发，同时维护应用程序的模型视图架构。Spring框架具有丰富的功能，让我们简单地讨论这些功能。

1．控制反转：控制反转或IoC是用于将服务或组件连接到应用程序的技术之一。 IoC是“一种软件设计模式和一组相关的编程技术，其中与传统交互模式相比，系统的控制流被反转。”在IoC而不是调用框架的应用程序中，它是调用的框架应用程序指定的组件。 IoC可以解释为“在运行时将所需资源或依赖项注入依赖资源”，也称为依赖注入。 org.springframework.beans.factory.BeanFactory是Spring IoC容器的实际表示，它负责包含和管理bean。 BeanFactory接口是Spring中的中央IoC容器接口。 bean只是一个由Spring IoC容器实例化和管理的对象。这些bean及其之间的依赖关系反映在容器使用的配置元数据中。

2.构造函数依赖注入：我们可以使用java类构造函数来加载bean值。 使用单个字段的构造函数定义java类。 Details.xml文件提供要传递给构造函数的值。 现在另一个java使用BeanFactory方法加载xml文件。 这使用xml文件在java文件的构造函数中加载值。 这用于将值传递给构造函数。

3. Setter依赖注入：我们为每个bean定义了getter和setter。 我们也可以使用setters方法设置bean中的值。 setters方法覆盖从bean加载的值。

4.接口：我们可以在spring中定义接口类。为了实现这一点，我们将导入java程序的接口。现在我们可以使用spring和xml在接口中定义的方法。

5.继承：一个ja​​va类可以像java程序一样获取另一个类的属性。它有三种子类型。1.摘要：宣称抽象的豆不能在泉水中遗传。2.父子：我们可以像父子一样定义层次结构。3.父-子–孙关系：在此我们可以为3个或更多个类定义层次结构。

6.自动装配：自动装配用于使用java文件映射属性名称，xml文件中的值。有四种类型可以集成它。 byName，byType，constructor，autodetect。如果没有定义任何内容，那么byName是默认值。

7. Bean的范围：Spring中定义的所有bean都具有四个值原型，会话，请求，单例，全局会话的范围。这用于控制bean的访问。

8.引用Bean：xml文件中的一个bean可以从另一个bean中分配值。这用于从一个bean读取值并分配给另一个bean。

# Spring的主要组件

在Spring里，我们也遵循MVC的原则。 它专为桌面和基于互联网的应用程序而设计。 Spring由三个核心协作组件组成。 1.控制器：处理导航逻辑并与业务逻辑2的服务层交互。模型：Controller和View之间的契约包含呈现控制器填充的视图所需的数据3.查看：呈现对请求的响应 从模型中提取数据。 弹簧MVC中的核心组件如下。

1. DispatcherServlet：这是spring的前端控制器实现。 Web.xml接收请求并将其传输到DispatchServlet。这是第一个与请求交互的控制器。它也被称为Servlet的实现。它控制应用程序的完整流程并导航应用程序流。

2.控制器：这是用户创建的处理请求的组件。它用它封装了导航逻辑。 Controller委托服务对象的服务。

3.视图：视图负责渲染输出。可以针对结果和观看设备，通信设备的不同类型的输出基础选择不同的视图。

4. ModelAndView：ModelAndView是spring框架的核心部分。它实现了应用程序的业务逻辑。它由控制器创建。它将视图与请求相关联。它存储业务逻辑和模型数据。控制器调用它并执行。执行时，它将返回数据和视图名称。

5. ViewResolver：输出的显示方式取决于从ModelAndView接收的结果。它用于将逻辑视图名称映射到实际视图实现。此部分标识并实现什么是输出媒体以及如何显示它。

6. HandlerMapping：DispatcherServlet用于将传入请求映射到各个控制器的策略接口。它标识请求并调用相应的处理程序来提供服务。处理程序将呼叫控制器。

# Spring的结构

Spring框架提供了一个用于构建Web应用程序的全功能MVC模块。 使用spring的可插拔MVC架构。 它可以使用多种视图技术配置Ex Java Server Pages，Velocity，Tiles，iText等.Spring MVC分离了控制器，模型对象，调度程序Servlet和处理程序对象的角色。 清晰的对象和控制器分离使它们更容易定制。 图01显示了执行流程的视图。

下图 3‑1显示了Spring模型的顺序图：

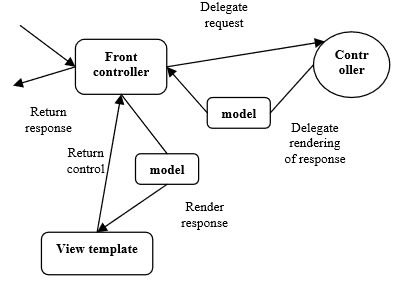


图 3‑1Spring模型的顺序图

在此，调度程序Servlet是应用程序的入口点。 一旦Dispatch Servlet获得服务请求，它就会决定处理程序。 所有处理程序都与Servlet一起映射。 处理程序将启动并将调用相应的控制器并将请求参数传递给它。

现在控制器开始运行，它包含业务逻辑，而ModelAndView与控制器相关联。 在执行时它会将ModelAndView返回到Dispatch Servlet。 此ModelAndView包含数据和视图名称。

Dispatcher Servlet从控制器获取ModelAndView。 它包含数据和视图名称。 Servlet将调用视图解析器。 视图解析器将标识要显示数据的视图的名称。 最后，它将向用户呈现相应且适当的格式的数据。

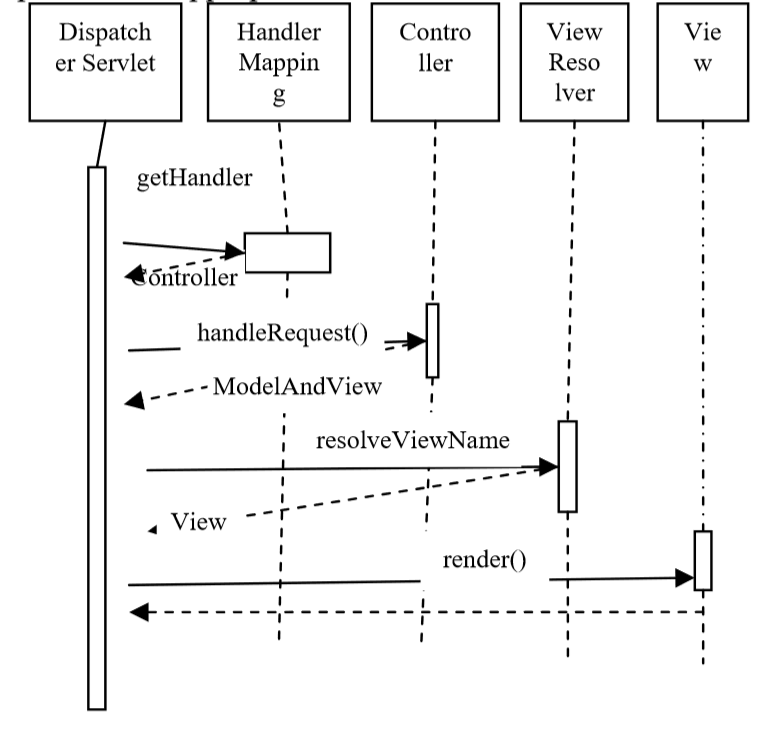


图 3‑2Spring框架中的应用顺序流程

# Spring和XML

Xml广泛用于spring框架中。 它简化了开发过程并节省了时间。 xml用于存储在执行应用程序期间使用的数据。

web.xml是应用程序中的入口点。 它会告诉你进一步的导航路径。 它加载应用程序上下文类，并告诉调度程序Servlet xml文件的名称。

<?xml version="1.0" encoding="UTF-8"?>

<web-appversion="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><listener-Class>

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 .xml将与服务器上的传入请求进行交互。 此文件将Servlet的名称定义为调度程序，ApplicationContext和index.jsp作为欢迎页面。 url模式定义为\*.\*表示它将是所有类型的传入请求。

**ApplicationContext.xml:** ApplicationContext构建在BeanFactory之上。 它提供了与Springs AOP功能，消息资源处理，事件传播的轻松集成。 BeanFactory提供配置框架和基本功能。 ApplicationContext为应用程序添加了增强功能。 在J2EE环境中构建应用程序时必须使用ApplicationContext。

<?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bean。 它还定义了引用类。

**Dispatcher-servlet.xml:-**

Spring的Web MVC框架是一个请求驱动的Web MVC框架，它围绕一个servlet设计，该servlet将请求分派给控制器并提供许多处理应用程序的功能。 DispatcherServlet与Spring ApplicationContext完全集成，允许您使用弹簧的功能。 Dispatcher Servlet是应用程序工作的中央控制单元。 它用于定义视图解析器，bean，处理程序及其应用程序的映射。

<?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>

# 如何运作快速应用程序开发

快速应用程序开发是很长一段时间以来业界的要求。 有许多开发工具可以帮助它。但逐渐地，技术和要求不断增加，因此需要能够处理应用程序规模不断扩大的工具和实际架构。工具与架构不同。 Spring是MVC中的架构，可以支持大型应用程序。在这种技术和架构中实现后，很容易继承应用程序而无需触及现有代码。 它使用xml文件帮助我们将新的映射，请求，java bean等添加到应用程序中。

# 结构优势

让我们看一下spring web MVC Framework可以为项目带来的一些架构优势。

1. Spring有效地组织了您的中间层对象，EJB不会影响它。 配置管理服务可以在任何架构层和任何运行时环境中使用。
2. Spring减少了组件的耦合。这是一个需要减少的主要问题，降低了可测试性和面向对象。
3. 明确的角色分离：Spring MVC很好地区分了构成这个Web框架的各个组件所扮演的角色。控制器，命令对象和评估器等所有组件的每个组件都扮演着不同的角色。
4. 可适应的控制器：如果您的应用程序不需要HTML表单，您可以编写一个更简单的Spring控制器版本，它需要表单控制器所需的所有额外组件。 Spring提供了几种类型的控制器，每种控制器都有不同的用途。
5. Spring通过在整个应用程序和项目中以一致的方式处理配置，消除了使用各种自定义属性文件格式的需要。
6. Spring通过降低编程到接口而不是类的成本来提供良好的编程实践。
7. 使用它构建的应用程序依赖于它的API。 spring应用程序中的大多数业务对象都不依赖于spring。
8. 使用Spring构建的应用程序非常容易进行单元测试。
9. Spring可以使EJB成为一种实现选择，而不是应用程序架构的决定因素。
10. 可以选择将业务接口实现为POJO或本地EJB，而不会影响调用代码。
11. Spring提供了适用于许多应用程序的EJB替代方案。它可以使用AOP在不使用EJB容器的情况下提供声明式事务管理。
12. Spring提供了一致的数据访问框架，无论是使用JDBC还是O / R映射，Hibernate。它在JDBC，JMS，JavaMail，JNDI和许多API等领域提供了一致且简单的编程模型，使其成为理想的架构。
13. Spring是使用POJO构建应用程序的框架。它还隐藏了开发人员的复杂性。
14. 在使用JDBC时它解决了连接泄漏的问题，我们只需编写必要的SQL，它也解决了从数据库返回的错误问题。

# 结论

Spring WEB mvc框架是一个框架，它为RAD环境中的应用程序提供环境。 在此框架中，我们可以依赖应用程序的一致性，性能和可靠性。 由于这是一个开源环境，因此建议开发人员在大型Web应用程序环境中继续使用此技术。

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

**附英文原文：**

图片包含 文字

描述已自动生成