

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/50281770>

Spring Web MVC Framework for rapid open source J2EE application development: a case

Article · July 2010

Source: DOAJ

CITATIONS

7

READS

1,456

3 authors, including:



Praveen Gupta

Poornima Group of Colleges

6 PUBLICATIONS 15 CITATIONS

[SEE PROFILE](#)



M. C. Govil

Malaviya National Institute of Technology Jaipur

50 PUBLICATIONS 129 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Underwater security [View project](#)



Reconfigurable Antennas for Wireless Communication Systems [View project](#)

All content following this page was uploaded by **Praveen Gupta** on 20 April 2015.

The user has requested enhancement of the downloaded file.

Spring Web MVC Framework for rapid open source J2EE application development: a case study

Praveen Gupta

Research Scholar, Singhania University
Pacheri Bari, Rajasthan, India

Prof. M.C. Govil

Govt. Mahila Engineering College
Ajmer, Rajasthan, India

Abstract— Today it is the highly competitive for the development of Web application, it is the need of the time to develop the application accurately, economically, and efficiently. We are interested to increase productivity and decrease complexity. This has been an underlying theme in a movement to change the way programmers approach developing Java 2 Platform, Enterprise Edition (J2EE) Web applications. Our focus is how to create J2EE-compliant software without using Enterprise Java Beans (EJB). The one of the best alternative is the Spring framework, which provides less services but it is much less intrusive than EJB. The driving force behind this shift is the need for greater productivity and reduced complexity in the area of Web application software development and implementation. In this paper, we briefly describe spring underlying architecture and present a case study using Spring web MVC Framework.

Index Terma: MVC, Spring, XML

I. INTRODUCTION

Web is the very complex issues these days. Since the desire of the companies and organizations are increasing so the complexity and the performance of the web programming matters. Complexity with the different types of communication devices is increasing. The business is demanding applications using the web and many communication devices. So with the increase load of the data on the internet we have to take care of the architecture issue. Let us discuss how it works fast using spring web mvc framework the rapid application development while maintaining the Model View Architecture of the application.

Spring frameworks comes with rich set of features, let us discuss these features in brief.

1 Inversion Of Control: Inversion of Control or IoC is one of the techniques used to wire services or components to an

application program. The IoC is "A software design pattern and set of associated programming techniques in which the flow of control of a system is inverted in comparison to the traditional interaction mode." In IoC instead of an application calling the framework, it is the framework that calls the components specified by the application. The IoC can be explained as "Injection of required resources or dependency at run-time into the dependent resource" which is also known as Dependency Injection. The `org.springframework.beans.factory.BeanFactory` is the actual representation of the Spring IoC container which is responsible for containing and managing the beans. The `BeanFactory` interface is the central IoC container interface in Spring. A bean is simply an object that is instantiated and managed by a Spring IoC container. These beans and the dependencies between them are reflected in the configuration metadata used by a container.

2. Constructor Dependency Injection: we can use the java class constructor to load the bean values. A java Class is defined with a constructor of single field. `Details.xml` file provides the value to be passes to the constructor. Now another java loads the xml file using the `BeanFactory` Method. This uses the xml file to load values in the constructor of the java file. This is used to pass values to the constructor.

3. Setter Dependency Injection: With every bean we defined the getters and setters. We can also use setters method to set the values in the beans. setters method overrides the values loaded from the beans.

4. Interface: we can define the interface class in spring. To implement this we will import interface to the java program. Now we can use methods defined in interface using spring and xml.
5. Inheritance: One java class can acquired the properties of another class just like a java program. There are three sub types of it. 1. Abstract: Beans declared abstract cannot be inherited in the springs. 2. Parent Child: we can define hierarchy like parent child. 3. Parent - Child - Sub Child Relationship: in this we can define hierarchy for 3 or more classes.
6. Autowiring: Autowiring is used to map the property name, values in xml file with java file. There are four types to integrate it. byName, byType, constructor, autodetect. If nothing is defined about it then byName is the default.
7. Scope of Beans: All beans defined in spring are having scope of four values prototype, session, request, singleton, global-session. This is used to control the access of the beans.
8. Reference Beans: One bean in the xml file can be assigned values from the other bean. This is used to read values from one bean and assign to another bean.

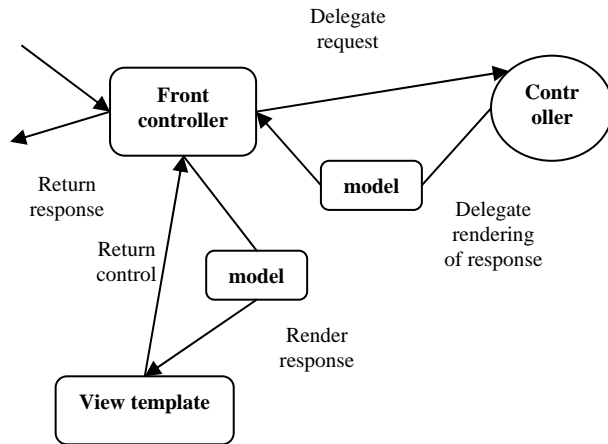
II. MAJOR SPRING COMPONENTS

In the spring we also follow the principals of the MVC. It has been designed more for the desktop and internet based applications. Spring consist of three core collaborating components. 1. Controller: Handles navigation logic and interacts with the Service tier for business logic 2. Model: The contract between the Controller and the View Contains the data needed to render the View Populated by the Controller 3. View: Renders the response to the request Pulls data from the model. Core components in the spring MVC are as follows.

1. DispatcherServlet: this is the spring's front controller implementation. Web.xml receives the request and transfer it to the DispatchServlet. This is the first controller which interacts to the requests. It is also known as implementation of the Servlet. It controls the complete flow of the application and navigates the flow of application.
2. Controller: this is the user created component for handling requests. It encapsulates the navigation logic with it. Controller delegates the services for the service object.
3. View: view is responsible for rendering output. Different views can be selected for the different types of output bases on the results and the viewing device, communication devices.
4. ModelAndView: ModelAndView is the core part of the spring framework. It implements the business logic of the application. It is created by the controller. It associates the view to the request. It stores the business logic and Model data. A controller calls it and it will execute. On execution it will return the data and name of view.
5. ViewResolver: How the output is to be displayed depends on the result received from ModelAndView. It is used to map logical view names to actual view implementations. This part identifies and implement what is the output media and how to display it.
6. HandlerMapping: Strategy interface used by DispatcherServlet for mapping incoming requests to individual Controllers. It identifies the request and calls the respective handler to provide the services. Handler will call to controller.

III. SPRING ARCHITECTURE

The Spring framework provides a full-featured MVC module for building Web applications. with spring's pluggable MVC architecture. It is configurable with multiple view technologies Ex Java Server Pages, Velocity, Tiles, iText etc. Spring MVC separates the roles of the controller, model object, dispatcher Servlet and the handler object. Clear separation of objects and controllers makes them easier to customize. The figure 01 shows the view of the execution flow.



The following figure 02 shows the sequence diagram of the spring model. In this the dispatcher Servlet is the entry point for the application. As soon the Dispatch Servlet get the request for the services and it will decides the handler. All handlers are mapped with the Servlet. Handler will come in action and will call the respective controller and the pass the request parameters to it.

Now controller comes in action, it contains business logic and a ModelAndView is associated with the controller. on execution it will return the ModelAndView to the Dispatch Servlet. This ModelAndView contains the data and view name.

Dispatcher Servlet gets the ModelAndView from the controller. It contains the data and view name. Servlet will call the view resolver. View resolver will identify the name of the view through which data is to be presented. Finally it will present the data to the respective and appropriate format to the user.

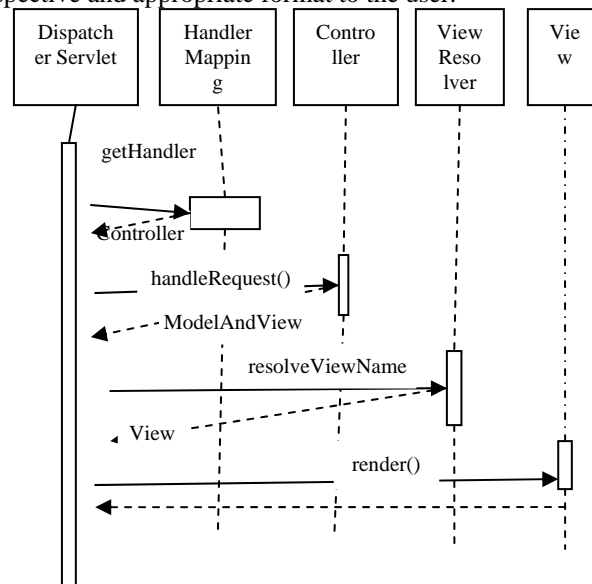


Figure 2: Sequence flow of application in the spring framework.

IV. SPRING AND XML

Xml is widely used in the spring framework. It simplify the development process and saves time. xml is used to store the data, which is used during the execution of application.

web.xml is the entry point in the application. It will tell you the further path of navigation. It loads the application context class and the tells the name of the dispatcher Servlet xml file.

```
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:schemaLocation="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><servlet-
class>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>
```

The above web.xml .xml will interact with the incoming request on the server. This file defines the name of the Servlet which is dispatcher, ApplicationContext and the index.jsp as welcome page. url pattern defined as *.* means it will all types of incoming request.

ApplicationContext.xml:- The ApplicationContext is build on top of the BeanFactory. It provides an easy integration with Springs AOP features, message resource handling, event propagation. The BeanFactory provides the configuration framework and basic functionality. ApplicationContext adds enhanced capabilities to application. While building applications in a J2EE-environment ApplicationContext must be used.

```
<?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>
```

The above file loads the bean of the SuperClass.java and SubClass.java. It also defines the reference class.

Dispatcher-servlet.xml:-

Spring's web MVC framework is a request driven web MVC framework, it is designed around a servlet that dispatches requests to controllers and provides much functionality for handling the applications. DispatcherServlet is completely integrated with the Spring ApplicationContext and allows you to use feature of springs.

Dispatcher Servlet is the central controlling unit for the working of the application. It is used to define the view resolver, beans, handlers and their mapping of the application.

```
<?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="package.name.DispatchController"></bean>
</beans>
```

V. HOW IT WORKS RAD

Rapid Application Development is the requirement of the industry since a long time. There are many development tools which have helped it. But gradually technology and the requirements goes on increasing so we need the tools and the in fact architecture which can handle the growing size of the application. Tools are different from the architecture. Spring is the architecture in MVC which can support the large applications. In this technology and architecture once implemented it is easy to inheritance the application without touching the existing code. It's use of the xml files helps us to add the new mappings, requests, java beans etc to the application.

VI. ARCHITECTURAL BENEFIT

let's look at some of the Architectural benefits spring web MVC Framework can bring to a project.

- Spring effectively organize your middle tier objects, EJB doesn't affect it. The configuration management services can be used in any architectural layer and in any runtime environment.
- The Spring Web MVC Framework is a robust, flexible, and well-designed framework for rapidly developing web applications using the MVC design pattern.
- Spring eliminate the proliferation of Singletons. This is a major problem, reducing testability and object orientation.
- Clear separation of roles: Spring MVC nicely separates the roles played by the various components that make up this web framework. All components like controllers, command objects, and valuators' each component plays a distinct role.
- Adaptable controllers: If your application does not require an HTML form, you can write a simpler version of a Spring controller that does need all the extra components required for form controllers. Spring provides several types of controllers, each serving a different purpose.
- Spring eliminate the need to use a variety of custom properties file formats, by handling configuration in a consistent way throughout applications and projects.
- Spring provides good programming practice by reducing the cost of programming to interfaces, rather than classes.

- Applications built with it depend on as few of its APIs. Most business objects in spring applications have no dependency on spring.
- Applications built using spring are very easy to unit test.
- Spring can make the use of EJB an implementation choice, rather than the determinant of application architecture.
- You can choose to implement business interfaces as POJOs or local EJBs without affecting calling code.
- Spring provide an alternative to EJB that's appropriate for many applications. It can use AOP to deliver declarative transaction management without using an EJB container.
- Spring provides a consistent framework for data access, whether using JDBC or an O/R mapping, Hibernate. It provides a consistent and simple programming model in areas like JDBC, JMS, JavaMail, JNDI and many APIs which makes it an ideal architectural.
- This is the framework which builds applications using POJOs. It also conceals complexity from the developer.
- While using JDBC it solve the problem of connection leak, we need to write only necessary SQL, it also solves the problems of error returned from database.

VII. CONCLUSION

Spring WEB mvc framework is an framework which provides the environment for the application in the RAD environment. In this framework we can rely for the consistency, performance and reliability of the application. Since this is an open source environment so it's recommended for the developers to go ahead with this technology for the large size of web application environment.

REFERENCES

- [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 Web-based 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

AUTHORS PROFILE

Praveen Gupta has received his Bachelor of Engineering in Electronics Engineering from Nagpur University, Nagpur, India, and Master in Technology in Information Technology from Punjabi University, India. He is working as Technical Leader in an software company based at Navi Mumbai, India. He is a member of various Technical Societies viz. Association of Computer Electronics and Electrical Engineers (ACEEE), He is a research scholar and pursuing his PhD at Singhania University , Pachari Bari, Rajasthan, India. His main research interests include: MVC, Java Design Patterns and Frameworks.

Prof. M.C. Govil has received Doctorate degree from IIT. He is working as professor and principal at govt women engineering college, Ajmer, India. He is having more than 20 years of experience in education and research. He is member of various Societies, he has published many papers. His research interest includes communication, web technologies, software engineering etc.