IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

3rd-International Conference on Muti-Disciplinary Application & Research Technologies (ICMART-2024)

Geetanjali Institute of Technical Studies

Vol. 11, Special Issue 2, May 2024

Web Applications Development using Struts 2 based Approach

Kamlesh Tiwari

Lecturer, CTAE, MPUAT, Udaipur, India

Abstract: This paper describes a step-by-step method for creating a Java application with the Struts Framework. The MVC-based web application is developed using the Struts framework. A key design pattern for separating business logic from user interface logic is the model view controller, or MVC. A sophisticated, expandable framework for corporate Java application development is called a strut. The framework is made to make the whole development cycle—from creating to delivering to maintaining apps—more efficient. Our study demonstrates that, in comparison to utilizing a single framework, designing applications employing MVC ideas is easier when many frameworks are used. More research articles that use a quick and innovative method to create web architecture instead of using a framework have been offered in recent trends.

Keywords*:* Java, Framework, Struts1, Struts2, MVC, ONGL, Web, Interceptor, POJO

I. INTRODUCTION

These days, the web is a highly complicated topic. The complexity and performance of web programming are important because businesses and organizations are becoming more and more demanding. The variety of communication devices is becoming more complex [1]. Many businesses utilize frameworks to make the development of their apps easier because the business is demanding applications that leverage the web and numerous communication devices. Web apps are now required by businesses to promote their brands, so application architecture must be carefully considered during creation. A framework is a collection of tools that developers can use to create apps [2]. Thus, we must address the architecture issue in light of the growing volume of data on the internet [1]. Let's take a quick look at the struts2 framework study. J2EE includes Struts. When establishing application programs using J2EE, designers and programmers must disperse the function in each discreteness of the server end, even if J2EE simplifies the foundation of enterprise level application programs [3]. In May 2000, Craig McClanahan of the Apache Foundation designed the struts framework, and in June 2001, struts 1.0 was made available. Struts 2.3.15.1 GA, released on July 16, 2013, is the most stable release. The MVC (Model View Controller) and POJO frameworks are developed with the Struts2 framework.

II. MODEL-VIEW-CONTROLLER

The software design pattern known as Model View Controller, or MVC as it is commonly known, is used to create web applications. The three components of a Model View Controller pattern are as follows: The model is the lowest level of the pattern and is in charge of data maintenance. The View function is in charge of presenting the user with all or some of the data. The Controller is a piece of software code that manages how the Model and View communicate with each other. Because it allows for the separation of responsibilities and isolates the application logic from the user interface layer, MVC is widely used. In this case, the Controller gets all application requests and collaborates with the Model to prepare any data required by the View. The Controller's prepared data is subsequently used by the View.

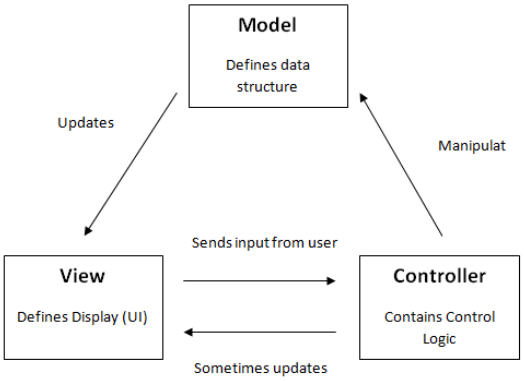

Fig. 1 MVC Architecture

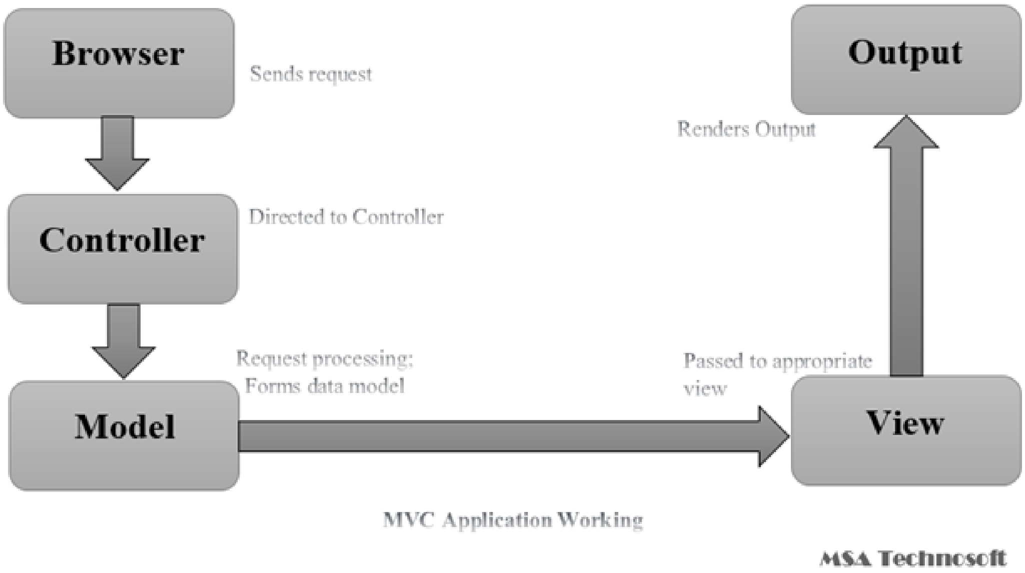

Fig. 2 Framework Architecture

III. DISADVANTAGE

One drawback is that we have to create the controller code ourselves. We must recompile the class and redeploy the application if we make changes to the controller code. Consider using Struts2 to get around these issues. Struts2 is the result of combining Struts1 and the Open Symphony web work framework. Struts2=Struts1+webwork.

WebWork: A next-generation web application framework with an action and request oriented architecture is being developed by the Apache Struts WebWork. Rickard Oberg originally developed WebWork to enhance the concepts and features of the Struts framework. Through a lengthy sequence of releases since then, the Web Work community has continued to expand and improve the framework's capabilities. Struts2 adds sophisticated Java 5 capabilities, Ajax, and templating to its strong WebWork base. Although the WebWork and Struts Action codebases have overlapped up until now, recent efforts to streamline the web framework market have motivated both groups to collaborate.

Struts1:Based on the Service to Worker paradigm, Struts is a framework that creates a strong and adaptable controller. The main benefits of struts are: Flexibility in integration: The architecture of Struts offers flexibility in terms of model and view selection. The concept of plug-ins forms the basis of the perspective [5]. An open-source framework called Struts1 uses an MVC (Model, View, Controller) architecture to augment the Java Servlet API. With the help of it, you can develop web apps that are flexible, expandable, and maintainable using common technologies like JavaBeans, JSP pages, resource bundles, and XML.

Disadvantage: Action classes must extend an abstract base class in order to use Struts 1. Using abstract classes rather than interfaces when developing is a prevalent issue with Struts 1.

Currently, Struts2 offers support for AJAX, validation, integration with the Hibernate and Spring frameworks, support for several result kinds including Velocity and JSP, and support for POJO-based actions. Numerous elements of a Struts2 are absent from Struts1. The struts2 framework's key characteristics are listed below:

1. An Action interface is implemented by an Action class. A base Action Support class that implements frequently used interfaces is provided by Struts2. Any POJO object with an execute signature can be used as a Struts2 Action object; an Action interface is not required.

2. Actions in Struts2 are not connected to a container. The servlet contexts are typically represented as straightforward Maps, which makes it possible to test Actions separately. If necessary, Struts2 Actions can still access the initial request and response. But other architectural features lessen or do away with the necessity of directly accessing the HTTP Servlet Request or HTTP Servlet Response.

3. Struts2 does not require a second input object because it uses Action attributes as input properties. Rich object types with their own set of properties could be the input properties. Through the taglibs, the Action properties are accessible from the webpage. POJO form objects, POJO Actions, and the Action Form design are all supported by Struts2.

Input/output objects can be of any rich object type, such as business or domain objects. It is easier to use taglib references to POJO input objects with the Model Driven functionality.

4. Through the use of interceptor stacks, Struts2 facilitates the creation of distinct lifecycles for each Action. As required, custom stacks can be made and utilized with various Actions.

5. Thread safety is not a problem with Struts 2 because Action objects are instantiated for every request. Each request that a servlet container receives produces a large number of throw-away objects; adding one more object has no negative effect on garbage collection or performance.

6. Testing Struts2 Actions involves instantiating the Action, configuring its properties, and calling its methods. The support for Dependency Injection also streamlines testing.

7. In addition to supporting JSTL, Struts2 also supports "Object Graph Notation Language," a more potent and adaptable expression language (OGNL).

8. Struts2 converts primitives, common object types, and Basic using converters and OGNL for type conversion.

9. Struts2 permits manual validation, which is carried out with the XWork Validation framework and the validate function. Using the validations set for the properties class type and validation context, the Xwork Validation Framework enables chaining of validations into sub-properties.

10. Value Stack technology is used by Struts2 to provide taglibs with access to the values without attaching the view to the object it is presenting. With multiple property types but the same property name, we can reuse views across a variety of kinds thanks to the Value Stack technique.

Struts 2 architecture: Let's attempt to comprehend the Struts2 application's standard architecture, as shown in Figure 1.3.

1. The activity is requested by the user.
2. The container obtains the controller's class name by mapping the request in the web.xml file.
3. The controller (Struts Prepare and Execute Filter or Filter Dispatcher) is called by the container. It is now Struts Prepare and Execute Filter since struts2.1. It was Filter Dispatcher prior to 2.1.
4. The Action Mapper provides the Controller with the action's information.
5. The Action Proxy is called by the Controller.
6. The configuration manager, which obtains its information from the struts.xml file, provides the action proxy with the action and interceptor stack details.
7. The request is sent to the Action Invocation by the Action Proxy.
8. Every interceptor and action are called upon by the action invocation.
9. A conclusion is reached.
10. The Action Invocation receives the result back.
11. A Servlet Http Response is produced.
12. The user receives a response.

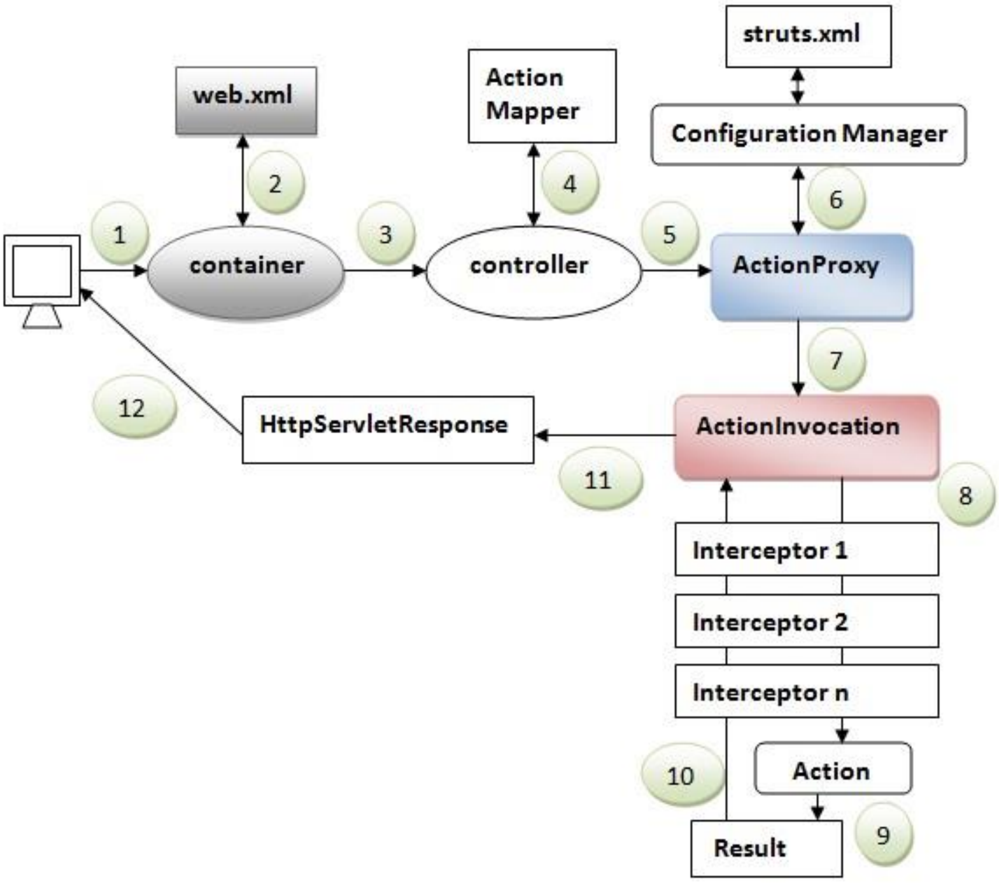

Fig. 3 Architecture of struts 2

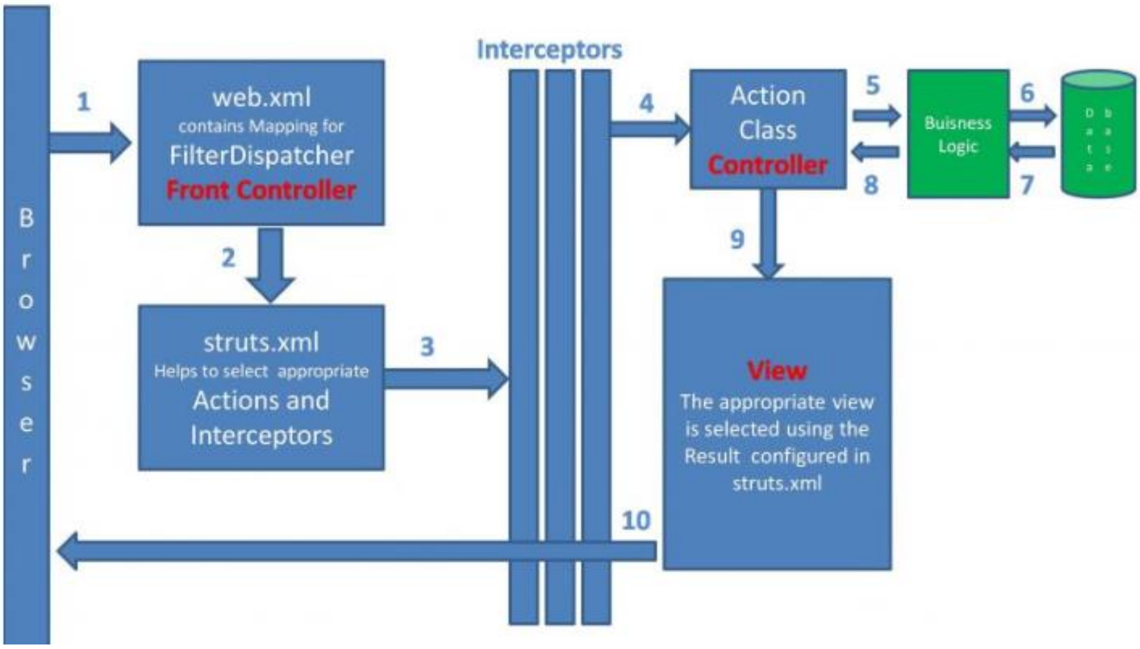

Fig. 4 Work flow of struts 2

When an HTTP request is sent by the client browser, the following things take place:

The request is received by the Action Servlet.

1. The browser sends the request to the server first. After the web.xml file has been loaded by the server, the request is routed to the FilterDispatcher if the request pattern matches. The Front Controller in Struts 2 is the FilterDispatcher.
2. The relevant action class to be run is determined by looking at the request url and its mapping in the struts.xml file.
3. The request passes through the interceptors prior to the Action class being executed.
4. The action method is executed of the action class (controller).
5. The business logic function is called by the action class.
6. The class of business logic is work on database.
7. Data fetched from database by Bushinesslogic class.
8. Processed data is sent back to the Action class.
9. The view to be produced is identified by the controller.
10. The interceptors are run one more before the response is created.

IV. IMPLEMENTATION

This section of the study discusses using Tomcat, an open-source servlet container created by the Apache Software Foundation (ASF), to implement the model Driven interceptor of Struts 2 [5]. Another model object is set as the default object of the value stack by the model-driven interceptor.

The action object is the value stack's default object by default. You must implement the Model Driven interface in your action class and override the obtain Model() function in order to use the Model Driven interceptor. By default, it can be found on the default stack. Thus, it is not necessary for you to state it clearly. For this approach, some of following files we require are:

struts.xml:
For this kind of web application, the struts.xml file needs to be defined. This file describes the package name and class configuration and applications in jsp files. The code following should be used to configure this:

```
<?xmlversion="1.0"encoding="UTF-8"?>
<!DOCTYPEstrutsPUBLIC
"-//Apache Software Foundation//DTD Struts Configuration 2.5//EN"
"http://struts.apache.org/dtds/struts-2.5.dtd">
<struts>
<packagename="default"namespace="/"extends="struts-default">
<actionname="getMessage"class="MyActionFile">
<resultname="success">/success.jsp</result>
<resultname="failure">/error.jsp</result>
</action>
</package>
</struts>
```

web.xml: Struts defines its Servlet Filter class, FilterDispatcher, in the web.xml file. This class manages all requests and initializes the Struts framework. Initialization parameters that determine what extra configuration files load and how the framework operates may be included in this filter.

```
<?xmlversion="1.0"encoding="UTF-8"?>
<web-appxmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="https://jakarta.ee/xml/ns/jakartaee"xmlns:web="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee                    https://jakarta.ee/xml/ns/jakartaee/web-
app_5_0.xsd"id="WebApp_ID"version="5.0">
<display-name>Demo</display-name>
<welcome-file-list>
<welcome-file>success.jsp</welcome-file>
</welcome-file-list>
<filter>
<filter-name>struts2</filter-name>
<filter-class>org.apache.struts2.dispatcher.filter.StrutsPrepareAndExecuteFilter</filter-class>
</filter>
<filter-mapping>
```

```
<filter-name>struts2</filter-name>
<url-pattern>/*</url-pattern>
</filter-mapping>
</web-app>
```

success.jsp:

```
<%@pagelanguage="java"contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPEhtml>
<html>
<head>
<metacharset="ISO-8859-1">
<title>Welcome</title>
</head>
<body>
<p>Welcome to my first application using Sturts 2 Framework</p>
</body>
</html>
```

error.jsp:

```
<%@pagelanguage="java"contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPEhtml>
<html>
<head>
<metacharset="ISO-8859-1">
<title>Error Page</title>
</head>
<body>
<p>Error Generated</p>
</body>
</html>
```

MyActionFile.java: The Struts 2 ModelDriven components are available on this page.

```
publicclassMyActionFile {
        public String execute() {
        System.out.println("Hello Everyone");
                return"success";
                }
}
```

V. FUTURE WORK

The greatest way to entice academic and research scholars to work is through open technology. The MVC-based architecture is best implemented with the Struts framework. Struts are a potent technology for business data that allow applications to access logic across platforms and from any source. The development of an enterprise application built on top of Spring, Hibernate, and Struts is the next task for this report.

VI. CONCLUSION

The solution to the data expression challenge has been suggested in this paper. It is capable of providing efficient business data protection based on the development architecture framework made up of Struts. Additionally, by dividing the data into several layers and separating the business logic from the data expression, it helps with higher level debugging and maintenance.

A large-scale corporation must be informational by necessity due to the development of the web. The Struts provide special emphasis on business logic control. Additionally, it will give businesses and programmers access to new markets and commercial prospects. Using this framework is advised for improved performance.

REFERENCES

[1] Garg R, Sood Y, Kottana B, et al. A Framework Based Approach for the Development of Web Based Applications. World Comp. Sci. Info. Technol. J. (WCSIT) 2011; 1(1): 1–4p.

[2] Meiyu Fang, Design and Implement of a Web Examination System Using Struts and EJB, Seventh International Conference on Web-based Learning 2008, 10.1109/ICWL.2008.25, ISBN:978-0-7695-3518-0.

[3] Wang Y, Guo C, Song L. Architecture of E-Commerce Systems Based on J2EE and MVC Pattern. International Conference on Management of e-Commerce and e Government, IEEE Computer Society 2009.

[4] J. Wojciechowski; B. Sakowicz; K. Dura; A. Napieralski, MVC model, struts framework and file upload issues in web applications based on J2EE platform, Proceedings of the International Conference Modern Problems of Radio Engineering, Telecommunications and Computer Science, 2004. ISBN:966-553-380-0

[5] Dipak Kumar Sahoo, MuraliSagar Varma Sagi, Bhupinder Singh, Pradeep Bhargava, MVC model, Effect of Detailing of Web Reinforcement on the Behavior of Bottle-shaped Struts, 2010 年 8 卷 3 号 p. 303-314.