Java Web Frameworks Which One to Choose?

Mohamadou Nassourou
Department of Computer Philology & Modern German Literature
University of Würzburg Am Hubland D - 97074 Würzburg
mohamadou.nassourou@uni-wuerzburg.de

Abstract

This article discusses web frameworks that are available to a software developer in Java language. It introduces MVC paradigm and some frameworks that implement it. The article presents an overview of Struts, Spring MVC, JSF Frameworks, as well as guidelines for selecting one of them as development environment.

1. Introduction

Over the last decade, the number of Java Web Frameworks has considerably increased. There are basically two types of Java Web Frameworks: component oriented frameworks and action based ones.
Action frameworks are mainly focussing on request/response processing. Action frameworks are very procedural with little reusability of code/components.
Component frameworks focus on object oriented web design. They do not concentrate on request/response processing.

There exist several actions frameworks among them Struts and Spring MVC that I am going to present. I will introduce Java Server Faces (JSF) which is a component framework.

All the frameworks that will be presented follow Model-View-Controller design pattern.

2. Model-View-Controller (MVC)

Model-View-Controller design pattern helps developers to better organise their program's code. In fact it provides a way of separating user interface i.e View from the business logic i.e Model. A Controller is responsible for invoking appropriate pages according to user's request. It determines also what business logic to call for a given request. Practically JSP pages represent the view and servlets the controller.

3. Struts

Struts is an open source framework created by Apache Jakarta [3, 6] for building web applications using Model View Controller (MVC) design pattern.

There are two versions of Struts: Struts1, and Struts2. Struts2 is an evolution of the legacy Struts1 framework.

A typical Struts application is made of:

- JSP pages containing Struts custom tags to display the user interface
- An action servlet that acts as the controller which forwards requests to action classes and selecting JSP pages to display
- ActionForm bean classes that populate and cache data on the forms
- Action classes that process user input as well as providing data for JSP pages
- An XML configuration file that defines the invocation process between URLs, action classes, form classes, and JSP pages. It provides the navigation rules.
- Resource files known as property files contain text strings for the application

3.1 Struts Development Environment

Following softwares are required:

- JDK (Java SE Development Kit)
- Apache Struts JAR files
- Application Server (Tomcat, JBoss or any standard application server having servlet engine)
- Integrated Development Environment i.e Eclipse

4. Spring MVC

The Spring Framework [1] is an open source application framework for Java platform and .NET Framework. Spring MVC is part of the Spring Framework. Like Struts, Spring MVC is a request-based framework. It offers the possibility to choose from several different view layers (JSP, Velocity, etc) because it is not tightly coupled with Servlets or Jsp to render the View to the Clients

A typical Spring MVC application consists of:

- A dispatcher servlet, which will intercept incoming requests.
- A configuration file which directs a web container to send requests to a dispatcher servlet
- Controllers that execute a business logic and route to relevant Views
- url mapping that instructs a dispatcher servlet how to map URLs to controllers
- view resolver that watches out which controller has completed and how to get to the view

4.1 Spring MVC Development Environment

- JDK (Java SE Development Kit)
- Spring distribution
- Application Server (Tomcat, JBoss or any standard application server having servlet engine)
- Integrated Development Environment i.e Eclipse
- Sysdeo Tomcat plugin

5. Java Server Faces (JSF)

JSF [5] was developed using Model-View-Controller paradigm. It is driven by Java Community Process (JCP) and has become a standard. JSF provides a set of pre-assembled User Interface (UI). This facilitates the use of pre-coded complex components. JSF can handle events. It has component model which enables the plugin of third-party components. JSF helps building web applications that run on a server and render the user interface back to the client.

A typical JSF application comprises:

- JavaBeans components containing application functionality and data
- Event listeners
- JSP pages
- Server-side helper classes
- A custom tag library for rendering UI components
- A custom tag library for representing event handlers and validators
- UI components made of stateful objects on the server
- Validators, event handlers, and navigation handlers.
- Configuration file for configuring application resources

5.1 JSF Development Environment

Following softwares are required:

- JDK (Java SE Development Kit)
- JSF distribution
- Application Server (Tomcat, JBoss or any standard application server having servlet engine)
- Integrated Development Environment i.e Eclipse

6. Choosing a Java Web Framework

After having presented the most popular Web Frameworks that exist for Java developers, I think one should decide which framework to use for one's projects. It would be unrealistic and not professional to select a Framework because it is popular or over advertised. The selection must be based on the project to be implemented. There is no framework that will be best for every project.

As guidelines I would recommand to answer the following questions first:

a) What programming paradigm is going to be used?

b) Does one need a component or an action based framework?

c) What is the size of the project?

d) What is the deadline to deliver the result?

e) Are the qualified developers available?

f) What would be the expectation of the client?

If the above questions are successfully answered, then the possibility of making a proper choice could be achieved. However the ultimate decision comes from the project owner.

7. Conclusion

In this article MVC design pattern has been introduced. Then the most popular Java Web Frameworks such as Struts, Spring MVC and JSF have been presented. The typical structure of each one has been mentioned as well as their required development environments.

Finally some guidelines for choosing one of them have been given.

As mentioned earlier the selection must be realistic and based on project specifications.

8. References

[1] http://en.wikipedia.org/wiki/Spring_MVC
[2] http://java-source.net/open-source/web-frameworks
[3] http://en.wikipedia.org/wiki/Apache_Struts
[4] http://en.wikipedia.org/wiki/Spring_Framework
[5] http://en.wikipedia.org/wiki/JavaServer_Faces
[6] http://jakarta.apache.org/