INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT (IJSREM)
VOLUME: 06 ISSUE: 07 | JULY - 2022 IMPACT FACTOR: 7.185 ISSN: 2582-3930

A Comprehensive Comparison Study on Spring Framework

Mohan Sharma V , Dr.S Anupama Kumar

***Abstract*—**In this research paper, I have tried to explore the way of building the web application with the basic needs of the user and which seems to be easy to build using spring framework. The major focus is on building web applications using the spring framework and its supported frameworks and supported software and programming languages and scripting language

Keywords—Spring framework, Java, MVC.

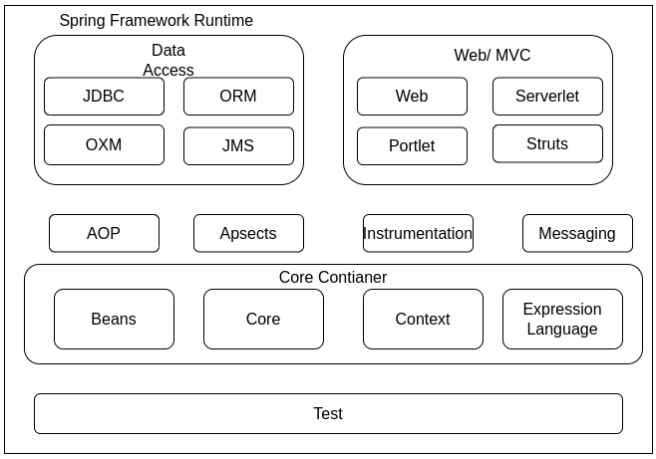
INTRODUCTION

The software requires continuous updates and the software is built feature by a feature which is known as the Agile methodology. The Spring framework is very suitable for the agile methodology. The developers prefer the Spring framework because it is very easy to update the software and upgrade the software. The spring framework is a very lightweight framework. The spring framework uses objects to run the application

The Spring framework is used to build the enterprise tools. The Spring is free and open-source which makes it very easy to learn for all. The basic version of the spring framework is almost 2 MB only. The spring framework is based on the Java programming language. The Spring framework is very helpful for upscaling the application with time.

Spring contains all types of frameworks which can full fill all the requirements to build an application. The spring framework manages the application using the Plain Old Java Object (POJO) which makes managing the classes easy.

I. ARCHITECTURE



The above diagram is an architecture diagram of the Spring framework. The Spring architecture supports the Plain Old Java Objects (POJO). The object basically does not need the support of any technologies. The lesser dependence on the technologies better the framework

The Core Container layer is the layer that manages the complete objects in the project. The Spring IoC container creates the objects on the basis of the needs and destroys the object after performing the required functions. The Beans define the actual mapping of the objects. The dependency injection shall be managed by the spring Core. The expression language is the most powerful which is used for manipulating and querying the objects at runtime. The Context module builds the solid base provided by the Spring IoC container and takes the java classes inside it and processes them according to the configuration class or configuration XML file.

The Data Access/ Integration layer consists of JDBC, ORM, OXM, and JMS and Transaction modules. The JDBC is used to access the database using the Java Application Programming Interface. The ORM module is used to perform the Object Relational Mapping. The ORM module includes the JPA, Hibernate. The Advantage of the ORM module is that the java objects are directly linked to the database. The java programming language can only be used to query the database and there is no need of any query language to query the database.

The Web layer consists of Web, Web-Servlet, Web-Struts, and Web-Portlet modules. The web module is used to provide web-oriented feature integration. The web layer is used to manage more number of java files, XML files, and Front -end files together to build a web application. The web servlet module consists of Spring Model View Controller(MVC) which is mainly used to build web applications. The Web module is used to integrate other modules and frameworks together and build a complete project

The Spring's AOP and Instrumentation module is used to give the freedom for the developer to define method-interceptors and pointcuts to clearly implement the functionalities that should be implemented separately in a decoupled manner.

The Test module is used to integrate the testing features to the project built in the spring framework. The testing tools like JUnit and TestNG can be used to test the application built using the Spring framework. The Test module provides the mock objects to test the working of the applications

II. SPRING MVC MODEL

Spring MVC provides the option for the developer to build the application using the Model View Controller (MVC) architecture. The Spring MVC framework belongs to the web module. The components that are available in the Spring MVC are servlets, Java servlet Pages (JSP), JAva Benas, and Bussiness logic. The major configuration files in the Spring MVC Framework is web.xml and Springbeans.xml file. The major components of the Spring MVC framework are

Controller: The controller module is responsible for taking the user input or the input from the client and giving the input to the Model module. The Controller using used to match the URL on the client-side and redirect to the user based on the user inputs. The @Controller classes are used to define the controller class. The controller will recognize the GET or POST and serve the logic

Model: The Model module is the module that gets the data from the Controller module and queries the database according to the API logic in the Model module. The Model modules get data from the database and pass the information to the View module.

View: The View module provides a front-end View to the client. The view module is to use render pages according to the Model module request.

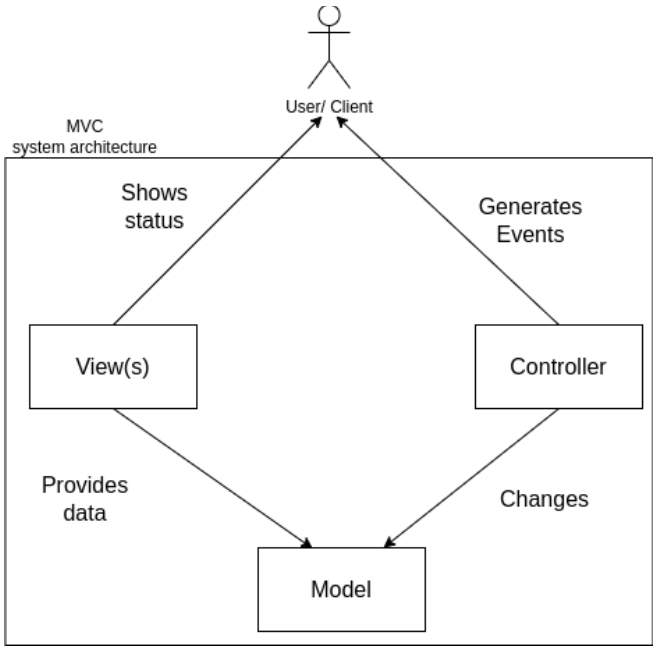


Figure3.1: Functionality of the layers in MVC architecture

figure 3.1 shows the Functionality of the layers in MVC architecture. what are the functionalities of each layer and the functions performed by each layers. The data given by one layer to another layer is also explained here.

Spring MVC backend architecture

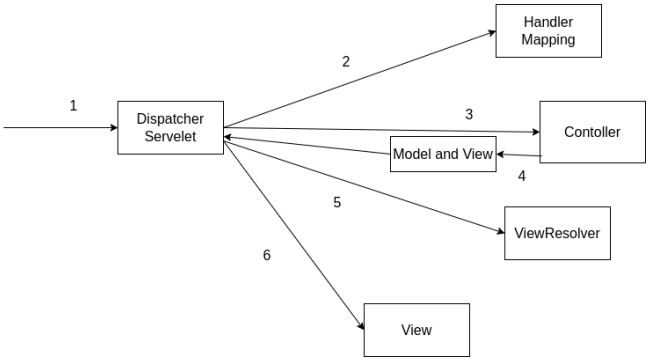
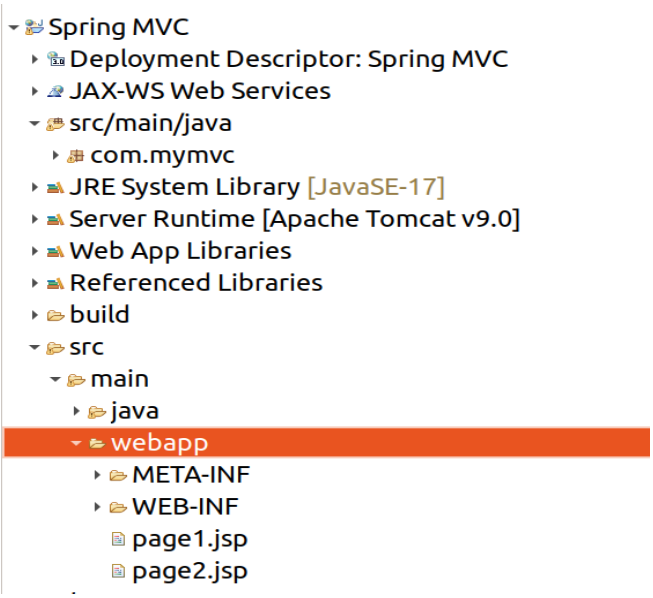


figure 3.2 Spring MVC backend architecture

The figure 3.2 Spring MVC backend architecture shows the importance of the Dispatcher servlet. The dispatcher servlet receives the request from the client and uses the Controller, Model, and Views modules according to the requirements. The Dispatcher Servlet takes the request to the Handler Mapping and processes it according. The Controller takes the input from the dispatcher servlet and processes it and the controller passes the data to the Model module. The Model module queries the database according to data given by the controller. The View module takes the request from the Model module and gives the front-end view or files to the client accordingly.

4. Implementation of the Spring framework

File Structure



Web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xmlns="http://java.sun.com/xml/ns/javaee"
	xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
	id="WebApp_ID" version="3.0">
	<display-name>SpringMVC</display-name>
	<servlet>
		<servlet-name>spring</servlet-name>
		<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
		<load-on-startup>1</load-on-startup>
	</servlet>
	<servlet-mapping>
		<servlet-name>spring</servlet-name>
		<url-pattern>/</url-pattern>
	</servlet-mapping>
</web-app>
```

2. Create the controller class

To create the controller class, we are using two annotations @Controller and @RequestMapping

Con.java

```
package com.mymvc;
import
org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import
org.springframework.web.bind.annotation.RequestBody;
import
org.springframework.web.bind.annotation.RequestMapping;
import
org.springframework.web.bind.annotation.RequestParam;

@Controller
public class Con {
@RequestMapping("/")
public String page1()
{
return "page1";
}
@RequestMapping("/page2")
public String page2(@RequestParam String nam,Model model)
{
	if(nam.length()>100)
	{
		String x="abdd";
		model.addAttribute("err",x);
		return "page1";

	}else {model.addAttribute("nam",nam);


	return "page2";
	}
}}
```

page1.jsp

The JSP file which is used to build the front end of the website

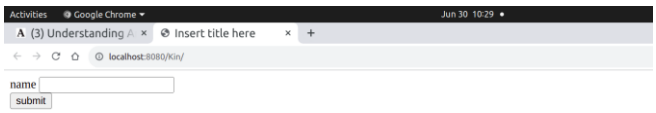
```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<form action="page2">
name <input type="text" name="nam">${err}
<br>
<button type="submit" > submit</button>
</form>
</body>
</html>
```

Pag2.jsp

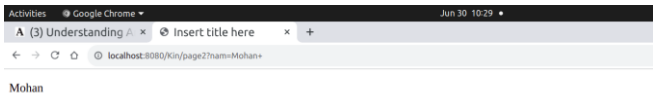
```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<p>
${nam}
</p>
</body>
</html>
```

Output:

Page1: default page



Page 2: redirected page from page1



The figure 3.2 shows the redirected page. The @RequestMapping shall redirect the user from the default page to this page

5. Problem And Key Alternatives

- The Spring framework are at times too complex
- The point of failure is sometimes high if the architecture is not designed properly
- The basics of the database are very important to complete the query
- Knowing the basics of the java is very important to make sure the product is built correctly

6.Key Alternatives

MERN Stack

The Mern stack is a javascript framework which is used build a complete web application using the React JS in the front-end, Express JS in middleware, Node JS in the back-end and MongoDB in the database

MEAN Stack

The Mean stack is a javascript framework which is used build a complete web application using the AngularJS in the front-end, Express JS in middleware, Node JS in the back-end and MongoDB in the database

PHP

The PHP can be used in the backend, HTML in the frontend and MySQL in the database part. Building the microservices is very hard in PHP

Conclusion

In this paper we have looked that spring architecture and the different modules available for integration in the spring framework. And the working of the spring framework with the use of all the modules. The modules which are used to build the web applications are known here. The Spring framework's MVC architecture and its working are known here. The MVC architecture application is easy to build and maintain. The implementation of the spring framework's MVC architecture isalso discussed above.

REFERENCES

[1] X. Blanc, MDA en action : Ingénierie logicielle guidée par les modèles. 1st edition, 270 pages, 2005.

[2]Spring Source WebSite (http://www.springsource.org/).

[3]SpringNetWebsite (http://www.springframework.net/).

[4] XML Metadata Interchange (XMI), version 2.1.1, December 2007, http://www.omg.org/spec/XMI/.

[5] Object Management Group (OMG). Model Driven Architecture. Retrieved December 12, 2013, from http://www.omg.org/mda/.

[6] Eclipse.org. ATLAS Transformation Language (ATL). http://www.eclipse.org/m2m/atl/.

[7] Mbarki, S. and Erramdani, "M. Toward automatic generation of mvc2 web applications," InfoComp - Journal of Computer Science, 7(4):84–91, December 2008.

[8] Mbarki, S. and Erramdani, "M. Model-driven transformations: From analysis to mvc 2 web model," International Review on Computers and Software (I.RE.CO.S.), 4(5):612–620, September 2009.

[9] Bezivin, J., Hammoudi, S., Lopes, D., Jouault, F., "Applying MDA approach for web service platform," In EDOC'04 preceedings of the 8th IEEE International Entreprise Distributed Object Computing Conference, pages 58-70, 2004.

[10] Czarnecki, K., Helsen, S., "Classification of Model Transformation Approaches," in online proceedings of the 2nd OOPSLA'03 Workshop on Generative Techniques in the Context of MDA. Anaheim, October, 2003.

[11] Gharavi, V., Mesbah, A., Deursen, A. V., "Modelling and Generating AJAX Applications: A Model-Driven Approach," Proceeding of the7th International Workshop on Web-Oriented Software Technologies, New York, USA (Page: 38, Year of publication: 2008, ISBN: 978-80-227-2899-7).

[12] Cong, X., Zhang, H., Zhou, D., Lu, P., Qin, L., "A Model-Driven Architecture Approach for Developing E-Learning Platform," Entertainment for Education. Digital Techniques and Systems Lecture Notes in Computer Science, Volume 6249/2010, 111-122, DOI: 10.1007/978-3-642-14533-9_12, 2010.

[13] Mbarki, Rahmouni, "MDA – Based Modeling and Transformation to Generate N – Tiers Web Models,"