J. Adv. Res. in Computer Graphics and Multimedia Technology
Volume 2, Issue 2 - 2020, Pg. No. 20-26
Peer Reviewed Journal

Article

Web Development Framework and Concepts Demystifed

Ankit Kelwa

Global Institute of Technology, Jaipur, Rajasthan, India.

INFO

E-mail Id:
ankit.kelwa@gmail.com

How to cite this article:
Kelwa A. Web Development Framework and Concepts Demystifed. *J Adv Res Comp Graph Multim Tech* 2020; 2(2): 20-26.

Date of Submission: 2020-09-20
Date of Acceptance: 2020-10-06

ABSTRACT

Every thing around us has started to appear on the internet that's what increasing the demand of the web development in the market. JAVA as a programming language has been glorified as being called as an evergreen immortal language. With the wide range and sub systems of JAVA in various field like stand-alone desktop applications, mobile applications, web development and many more. As the time changes new frameworks and concepts are used. The MVC design pattern which is a helpful pattern for the advance of web applications as it clearly distinguishes between model-view-controller. This improves the modification capabilities with the passage of time and combined with agile methodology it is a fast process. Also the company benifited as their cost of changing software reduces, users are happy with fast updates and developers without any burden of extra coding. How it supportive? This paper would help you know the answers to it.

Keywords: Frameworks, JAVA, Web development, Model View Controller

Introduction

W EB development is a critical section to look up for a industry and the selection of the programming language is more tougher. When JAVA arrived in the late 90's twentieth Century it was surrounded by numerous competitions like PHP, Python, Ruby, JAVAScript, Lua, and Haskell which all were introduced in same era. JAVA struggled to become a web interactive language, but ended up lagging behind to JAVAScript and other newer technologies.

However, JAVA amplified popularity as compared with its competition which made it utmost preferred for any kind of use intensification and enterprise backend growth. Now, it has been able to reach the newer heights with after the ongoing research for the packages which are able to perform operations of data manipulation and data interpretation which were not possible before.

Why JAVA?

JAVA is currently the most popular programming language among Para-academics and computer practitioners. Java was first established by James Gosling from Sun Microsystems in the 1990s. JAVA initially was developed to meet the needs of a language a computer written once and can be run on many systems different computers without meaningful code alterations. Most languages existing computers have different system migration limitations and not have cross platform support.

JAVA was created as a new language with implementation different. The Java language is an object-oriented language derived from C++ with many improvements. Now universities in various countries turned away from Pascal or C++ or Delphi then chose JAVA as a language for learning programming. As a programming language that many people like because its programming concepts are consistent with Object Orientation Theory as well safe to use, so it have several advantages over others as it is simple, Object Oriented, distributed, safe, neutral architecture, portable, interpreter, powerful, multithreading and dynamic.

Following are the distinctive features in JAVA making it worthy to be used more commonly.

Java Virtual Machine (JVM)

JVM is an imaginary machine (virtual) that works by resembling applications on a real machine. JVM provides hardware and platform specifications where compilation of JAVA code occurs. These specifications make JAVA based applications free from any platform because the compilation process is completed by the JVM and improve their portability. The program applications are created with files .java extension text. This compilation program generates one byte code file with a class extension or more.

Byte code is a series of instructions related to machine code instructions. The difference is that the machine code must be run on the computer system where the compilation is anticipated, while the byte code runs on the java interpreter which is available on all computer system platforms and operating systems. This helps out it to become a preferred programing language for desktop applications.

Garbage Collection

Many other programming languages allow a program to allocate memory at run time. However, after using these memory allocations, there must be a way to replace the memory blocks so other programs can use them. In C, C ++ and other languages, it is the programmer who is absolutely responsible for this. This can be difficult if the programmer neglects to restore the memory block so that it causes a situation known as memory leaks. The Java program performs garbage collection, which means the program does not need to delete objects that are no longer used. This ability reduces the burden of memory management by the programmer and reduces or eliminates the biggest source of errors found in languages that allow dynamic allocation.

Code Security

Code Security implemented in Java through the use of JAVA Runtime Environment (JRE). Java uses a 3-layer security model to protect the system from untrusted Java Code.

- First, class-loader handles loading Java classes into runtime interpreter. This process provides security by separating classes from local disk with classes taken from the network. This limits the Trojan application because the classes are from local disk that is loaded first.
- Second, the bytecode verifier reads the bytecode before it runs and ensures that the bytecode meets the basic rules of the Java language.
- Third, security management handles application-level security by controlling whether programs have the right to access resources such as file systems, network ports, external processes and systems windowing. After the whole process is finished, then the program code is executed.

MVC

Model-View-Controller (MVC) offers sustenance for rapid and parallel expansion .Developing web applications using the MVC modelit is possible that one developer work on the view while additional can work on the controller. This helps for easy implementation of the business logic of the webapplication. It surely benefits developers for completing the web application three times quicker compares with the uses that are developed using other development patterns. It is a design pattern for the application architecture, within this pattern divisions are established between what is the logic of the application, what are the presentation and the data, each of these parts grouping three different components.

Design Components

- Model represents the data that the application handles, typically they are the functions that support to insert, update, retrieve and delete information from a database.
- View is the information that is presented to the user, is the interface through which the user interacts with the system.
- Controller is the intermediary between the Model and the View, within this element is everything that the application can do, it is in charge of responding to the events that the user invokes.

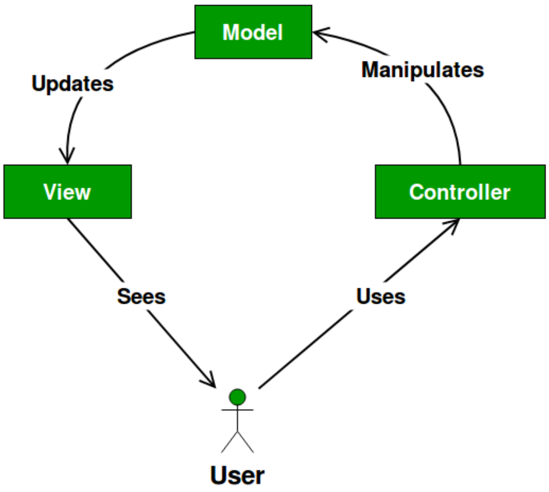

Figure 1.UML Diagram MVC Design[1]

Implementation of MVC

The MVC is an architecture that separates business logic, presentation and data. This definitely is similar to the J2EE architecture, with the help of components like Servlets, Scriptlets and JSP it is easy to perform operations. The flow of data is started from view layer, which then processed by the controller layer and then insertion of data further the apropiate rpely meassage is reverted.[Ref. Appendix A].

Advantages & Disadvantages of MVC

Despite of being a most preferred design pattern it has different interpretations and outcomes as per the user or programmer. So let us have a look oversome points worth mentioning into the advantages and disadvantages listed below.

Advantages of the MVC:

- Code separation into three components provides easy to maintain elements.
- It provides the facility to add multiple views of the same data or information.
- Facilitates adding new types of data as required by the application, since they are independent of the operation of the other components.
- It offers the ability to scale the application if required.
- In the case of projects where there are several developers, following common programming methods makes the code more understandable among these, one can continue the work of another.

Disadvantages of the MVC:

- Component separation adds inherent complexity to the system.
- The amount of files to maintain and develop increases considerably.
- The learning curve of the design pattern is high.

Frameworks

Spring MVC

Despite being a vintage outline (more than ten years ago from its first version), it is still at the top of the wave. After his Expanding to MVC (Controller View Model), spring continued to evolve into a large-scale JAVA framework for Internet applications, offering engineers a powerful tool for developing web applications as well as for security projects.[2]

Spring is not at number one for any reason. These are the most outstanding:

- Simplified injection of test data through the use of POJOs.
- Enhanced modularity, subsequent in better code readability.
- Flexible coupling amid the dissimilar modules.
- Dependency injection with flexible use.

Struts 2

Another of the main JAVA frameworks used by many of the most modern software engineers is the successor to Apache Struts 1, its number two version. This framework is a real find for developers building new web applications with JAVA EE. Apache has delivered to them an enormous set of tools to create business-oriented web applications optimizing the development process from start to finish and post-commissioning maintenance.

Pros and Cons

It is a luxury not to have to appliance the Action class since it is already incorporated. The code to configure repeats is no longer necessary because the interceptors take care of it. However, you might not go for this framework for one reason: many developers don't consider it agile enough to work with it.

Vaadin

This JAVA framework became one of the main and most popular for business application development. Using a Approach based on known components, takes away weight from developers when communicating changes made to the browser. A comprehensive set of user interface mechanisms, along with various widgets and controls, allows developers to create more than interesting applications in no time.

Pros and Cons

Doing the layout in JAVA, HTML or both, Vaadin allows linking data using the MVC or the MVP. Drag and drop, together with a large number of features, make it easy to create Single Page UIs with a JAVA application. On the downside, the slow user interface for mobile apps can be troubling since that Vaadin sends each event back to the server.

Other PHP Frameworks

Laravel

Laravel is one of the utmost general frameworks today, being used by a lot of developers. It is a rather young framework, since it was released in February 2012, but it has caught the attention of quite a few programmers since then. The latest version of Laravel is made up of several Composer packs. It integrates a good route management, authentication system, support for migrations from other environments and the Blade model management, among others things. Their template system, called Blade, is very light and has a quite elegant syntax, facilitating most of the tasks such as login actions, session control, managing cached pages and more. Laravel has a local development environment called Homestead, which is included in a Vagrant box.

Symfony

Symfony is one of the most flexible and scalable PHP frameworks that exist for the improvement of applications that use the view-controller or MVC model. There are many components for PHP that can be used such as Form Config for the organization of forms, Translation for the management of translations, Templating for the management of topics and Security, which as its name indicates is a security component. Like Laravel, it is also distributed through a series of modular packages using the Composer dependency manager.

Symfony deals of making the developing web applications is easier and less repetitive, in addition to making applications are easier to maintain. In general, the components of Symfony 2 are a series of PHP libraries that can be reused, useful for creating forms, configuring routes and authenticating users.

CodeIgniter

CodeIgniter is a free PHP framework that is maintained by EllisLab, also creators of Expression Engine, another paid PHP framework that incorporates quite a few CodeIgniter basics. It has a large number of documentation and a huge user communal. It is a framework that stands out for how light it is, since it only takes up 2MB of space.

There are many interesting CodeIgniter functionalities such as the fact that it is extremely easy to install, error handling, the integrated security system and the inclusion of an encryption system, as well as many other libraries included by default.

Due to its simplicity, its learning curve is very simple, requiring are relatively short time to master. It is an ideal framework for beginners.

Due to how easy it is to install, you will avoid many problems of conflicts or problems concerning the configuration of the server are present in other frameworks like Zend, orat the very least, they are the first time you use it. That is why it is possible to use it in shared hosting accounts without problem.

JAVA Scripts

ReactJS

ReactJS is undoubtedly the most popular has done between these last two years but. Why do people like it so much since in theory it is just a "library"? Here are the advantages and disadvantages of this "library for drawing user interfaces" as Facebook presents it on its official page.

Advantages:

- It works on the basis of "Components" which are pieces of code that can be reused multiple times throughout your application and thus keep the code much more organized and understandable.
- ReactJS uses a syntax that closely resembles the very co Known HTML, Facebook people call it "JSX" which has slight differences with HTML but at the end and at Cabo makes us very familiar when working with this powerful library.

Angular

Angular Maintained by Google, Angular is a giant that is well known and loved by its community; this framework helps us to create super complex and scalable Enterprise-level applications in a very clean way.

Advantages:

- Angular as in its previous versions it has the "two-way data binding" what which allows us to change the data of our application immediately, and these are reflected in the view, and vice versa, this is a characteristic of Angular and as we develop with this Framework, it is very easy for us to manipulate the rendered data.[3]
- TypeScript which is a JAVAScript superset that provides data typing to our variables, making our code much more robust and thus avoiding problems in the execution environment.

VueJS

VueJS uses, despite existence preserved by a large one such as are Google or Facebook not has nothing to envy to their consistent frameworks/ libraries and here I will elucidate the compensations and difficulties of using this framework.

Advantages:

- Easy to learn and syntax of objects well known by all developer JAVAScript also works on a component basis which makes it a very attractive framework
- Quite flexible and fast when rendering data on screen-from

Tools and other Factors

IDEs and Refactoring

Java had enabled and created the rich Integrated Development Environments (IDEs) as we know them. Don't get me wrong. IDEs had existed before for many different languages and they were written in many different languages, too. Yet IDEs, being big desktop UI applications with many plugins, had benefited immensely from the JAVA's combination of static typing, managed memory, dynamic code loading, and cross-platform feature. Modern IDEs, like IntelliJ IDEA and others based on IntelliJ platform, contain almost ten million lines of code and more than ten years of legacy. The big "IDE wars" of XXI Century between IDEA, Eclipse and Net beans were and are fought between JAVA-based desktop IDEs. This had created a feedback loop where JAVA IDEs had allowed scaling up development and, in turn, creating even more complex and feature-rich IDEs.

Around the turn of the Century, Martin Fowler published the Refactoring book on enlightening the design of prevailing code. It additional a novel word to the software designer's vocabulary. IDEs had wedged up and employed fully automated refactoring support, eventually transforming the way software was written to begin with. It used to be the case that you had to think through the details and the arrangement of your code in agonizing part in improvement, meanwhile a disappointment to foreknow the essential to extract a certain piece of logic into a function or a class, seriously chosen name or some other abstraction disappointment, would chief to expensive, droning and utterly non-fun rework later on. Not any longer. With IDE-supported refactoring you start writing your code in top-down fashion, presenting abstractions and rechristening them as you discover the need for them. I'd say that this was one of the greatest enhancements in software designer productivity of the XXI Century.

Java was exclusively located to reap the most welfares of automated refactoring by a pure chance. The innovative JAVA language design did not comprise any kind of macro system nor preprocessor since of the desire for straightforwardness and cross-platform feature. These conclusions turned out to be gold and made JAVA language extraordinarily well-matched to safe and computerized refactoring in IDE.

The sum is greater than the parts

It is a mixture of all those "right things", some of which I've revealed, that quickly moved JAVA to the prime spot amongst the programming languages. Other languages tried to duplicate JAVA's success, like Microsoft's C#, which congenital nearly all of JAVA's strengths. However, it originally locked its users to Microsoft platform, failing to distinguish the position of cross-platform feature that JAVA presented. That was an overpriced mistake, between some others, that took years to realize and had significantly undermined C#'s potential.

Acknowledgment

The authors wish to express deep gratitude to Mr. Sunil Gupta and Ms. Shradhanjali Sharma for making this happen. Also thanking Mr. Girraj Khandelwal for guidance.

Conclusion

Without a doubt, the inclusion of new technologies, libraries or enlargement tools application, allows developers/programmers know new useful mechanisms and agile to be able to develop systems capable of respect standards and patterns, without neglecting the tests, so that they allow, among other things, things the quality of the software. It is important that companies dedicated to software development guide your programmers to train them so that they allow you to use such tools as a viable alternative to emerging software with quality in a more agile and reliable way.

The software development is being revolutionalized and all kind of services are directing towards shaping the forthcoming of web technologies. With the latest improvements and updates the era of JAVA would be sustaining till a better alternative is found. There is nothing wrong to say that this language is gonna remain evergreen Long live JAVA!.

References

1. https://www.geeksforgeeks.org/mvc-design-pattern/
2. https://docs.spring.io/spring/docs/current/spring-framework-reference/
3. https://docs.angularjs.org/guide/databinding
4. https://www.javatpoint.com/MVC-in-jsp
5. Kelwa A. Author Graduating in 2020 from Global Institute of Technology, Jaipur with a good academic record. Completed summer internship from MetaCube Software, Jaipur. Have intrest in development of softwares by learning new technologies like Python & Ruby. Currently learning about artificial intelligence.

Appendix

Appendix A

Code Example[4], we have created 5 files:

- index.jsp a page that gets input from the user.
- ControllerServlet.java a servlet that acts as a controller.
- login-success.jsp and login-error.jsp files acts as view components.
- web.xml file for mapping the servlet.

File: index.jsp

```
<form action=”ControllerServlet” method=”post”>
Name:<input type=”text” name=”name”><br>
Password:<input type=”password” name=”password”><br>
<input type=”submit” value=”login”>
</form>
```

File: ControllerServlet

```
package com.javatpoint;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class ControllerServlet extends HttpServlet {
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
esponse.setContentType(“text/html”);
PrintWriter out=response.getWriter();
String name=request.getParameter(“name”);
String password=request.getParameter(“password”);
LoginBean bean=new LoginBean();
bean.setName(name);
bean.setPassword(password);
request.setAttribute(“bean”,bean);
boolean status=bean.validate();
if(status){
RequestDispatcher rd=request.getRequestDispatcher(“login-success.jsp”);
rd.forward(request, response);
}
else{
RequestDispatcher rd=request.getRequestDispatcher(“login-error.jsp”);
rd.forward(request, response);
}
}
@Override
protected void doGet(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
doPost(req, resp);
}
}
```

File: LoginBean.java

```
package com.javatpoint;
public class LoginBean {
private String name,password;
public String getName() {
return name;
}
public void setName(String name) {
this.name = name;
}
public String getPassword() {
return password;
}
public void setPassword(String password) {
this.password = password;
}
public boolean validate(){
if(password.equals(“admin”)){
return true;
}
else{
return false;
}
}
}
```

File: login-success.jsp

```
<%@page import=”com.javatpoint.LoginBean”%>
<p>You are successfully logged in!</p>
<%
LoginBean bean=(LoginBean)request.getAttribute(“bean”);
out.print(“Welcome, “+bean.getName());
%>
```

File: login-error.jsp

```
<p>Sorry! username or password error</p>
<%@ include file=”index.jsp” %>
```

File: 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” xmlns:web=”http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd”
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”>
<servlet>
<servlet-name>s1</servlet-name>
<servlet-class>com.javatpoint.ControllerServlet</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>s1</servlet-name>
<url-pattern>/ControllerServlet</url-pattern>
</servlet-mapping>
</web-app>
```