



Spring – MVC Framework

D dudhanevrushali28

Read

Discuss

Practice

Spring MVC Framework follows the Model-View-Controller design pattern. It is used to develop web applications. It works around DispatcherServlet. DispatcherServlet handles all the HTTP requests and responses. It dispatches the requests to handlers. It uses @Controller and @RequestMapping as default request handlers. The @Controller annotation defines that a particular class is a controller. @RequestMapping annotation maps web requests to Spring Controller methods. The terms model, view, and controller are as follows:

- **Model:** The Model encapsulates the application data.
- **View:** View renders the model data and also generates HTML output that the client's browser can interpret.
- **Controller:** The Controller processes the user requests and passes them to the view for rendering.

Spring MVC Framework works as follows:

1. All the incoming requests are intercepted by the DispatcherServlet that works as the front controller.
2. The DispatcherServlet then gets an entry of handler mapping from the XML file and forwards the request to the controller.
3. The object of ModelAndView is returned by the controller.
4. The DispatcherServlet checks the entry of the view resolver in the XML file and invokes the appropriate view component.

Advantages of Spring MVC Framework

- The container is used for the development and deployment of applications and uses a lightweight servlet.
- It enables rapid and parallel development.
- Development of the application becomes fast.



We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

- It is Easier to Debug because we have multiple levels in the application.

Disadvantages of Spring MVC Framework

- It has high complexity to develop the applications using this pattern.
- It is not suitable for small applications which affect the application's performance and design.

Create Your First Spring MVC Application

Consider the following example:

Step 0: Setup your project with maven use the required archetype to get the required folders directory and configure the server with your project.

Step 1: Load the spring jar files or add the dependencies if Maven is used. Add the following dependencies in pom.xml

pom.xml

XML

```
<project xmlns="http://maven.apache.org/POM/4.0.0" \
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
                      http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.javatpoint</groupId>
  <artifactId>SpringMVC</artifactId>
  <packaging>war</packaging>
  <version>0.0.1-SNAPSHOT</version>
  <name>SpringMVC Maven Webapp</name>
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#)

```

    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>3.8.1</version>
    <scope>test</scope>
</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-webmvc -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>5.1.1.RELEASE</version>
</dependency>

<!-- https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>servlet-api</artifactId>
    <version>3.0-alpha-1</version>
</dependency>

</dependencies>
<build>
    <finalName>SpringMVC</finalName>
</build>
</project>

```

Step 2: Create the Controller Class

HelloGeek.java

Java

```

@Controller
public class HelloGeek {
    @RequestMapping("/")
    public String display()
    {
        return "hello";
    }
}

```

Step 3: Provide the name of the controller in the web.xml file as follows:

DispatcherServlet is the front controller in Spring Web MVC. Incoming requests for the HTML file are forwarded to the DispatcherServlet.

web.xml

XML

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#)

```

<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"
  <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>

```

Step 4: We have to define the bean in a separate XML file. We have specified the view components in this file. It is located in the WEB-INF directory.

spring-servlet.xml

XML

```

<?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:context="http://www.springframework.org/schema/context"
  xmlns:mvc="http://www.springframework.org/schema/mvc"
  xsi:schemaLocation="
    http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context.xsd
    http://www.springframework.org/schema/mvc
    http://www.springframework.org/schema/mvc/spring-mvc.xsd">

  <!-- This element defines the base-package where
        DispatcherServlet will search the controller class. -->
  <context:component-scan base-package="com.geek" />

  <!--Provide support for conversion,
        formatting and also for validation -->
  <mvc:annotation-driven/>

</beans>

```

Step 5: Use JSP to display the message

index.jsp

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#)

HTML

```
<html>
<body>
```

```
<p>Spring MVC Tutorial!!</p>
```

```
</body>
```

Java Arrays Java Strings Java OOPs Java Collection Java 8 Tutorial **Courses @SALE** Java Multithreading

Step 6: Start the server and run the project. The output is displayed as follows:

Spring MVC Tutorial!!

Last Updated : 31 May, 2022

19

Similar Reads



Difference between Spring MVC and Spring Boot



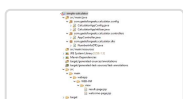
Create and Run Your First Spring MVC Controller in Eclipse/Spring Tool Suite



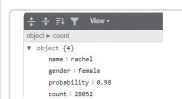
Spring Boot - Spring JDBC vs Spring Data JDBC



Difference Between Spring DAO vs Spring ORM vs Spring JDBC



Data Transfer Object (DTO) in Spring MVC with Example



Spring MVC - Get Probability of a Gender by Providing a Name using REST API



Spring - MVC Regular Expression Validation



Spring MVC using Java based configuration

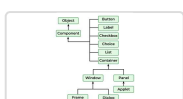


Spring MVC - Get Exchange Rate Values using REST API



Spring - MVC Form Drop-Down List

Related Tutorials



Java AWT Tutorial



Spring MVC Tutorial



Spring Tutorial



Spring Boot Tutorial

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#)



Java 8 Features - Complete Tutorial

[Previous](#)[Next](#)

Article Contributed By :

D

[dudhanevrushali28](#)
dudhanevrushali28
[Follow](#)

Vote for difficulty

Current difficulty : [Medium](#)

Easy

Normal

Medium

Hard

Expert

Improved By : [tyagiashwin193](#)

Article Tags : [Geeks-Premier-League-2022](#), [Java-Spring](#), [Java-Spring-MVC](#), [Picked](#), [Geeks Premier League](#), [Java](#)

Practice Tags : [Java](#)

[Improve Article](#)[Report Issue](#)



Company

About Us
Legal
Careers
In Media
Contact Us
Advertise with us

Languages

Python
Java
C++
PHP
GoLang
SQL
R Language
Android Tutorial

Explore

Job-A-Thon Hiring Challenge
Hack-A-Thon
GfG Weekly Contest
Offline Classes (Delhi/NCR)
DSA in JAVA/C++
Master System Design
Master CP

DSA Concepts

Data Structures
Arrays
Strings
Linked List
Algorithms
Searching
Sorting
Mathematical
Dynamic Programming

DSA Roadmaps

DSA for Beginners
Basic DSA Coding Problems
Complete Roadmap To Learn DSA
DSA for FrontEnd Developers
DSA with JavaScript
Top 100 DSA Interview Problems
All Cheat Sheets
DSA Roadmap by Sandeep Jain

Web Development

HTML
CSS
JavaScript
Bootstrap
ReactJS
AngularJS
NodeJS
Express.js
Lodash

Computer Science

GATE CS Notes
Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths

Python

Python Programming Examples
Django Tutorial
Python Projects
Python Tkinter
OpenCV Python Tutorial
Python Interview Question

Data Science & ML

Data Science With Python
Data Science For Beginner
Machine Learning Tutorial
Maths For Machine Learning
Pandas Tutorial
NumPy Tutorial
NLP Tutorial
Deep Learning Tutorial

DevOps

Git
AWS
Docker
Kubernetes
Azure
GCP

Competitive Programming

Top DSA for CP
Top 50 Tree Problems
Top 50 Graph Problems
Top 50 Array Problems

System Design

What is System Design
Monolithic and Distributed SD
Scalability in SD
Databases in SD

[Top 15 Websites for CP](#)[Top SD Interview Questions](#)

Interview Corner

[Company Wise Preparation](#)[Preparation for SDE](#)[Experienced Interviews](#)[Internship Interviews](#)[Competitive Programming](#)[Aptitude Preparation](#)

Commerce

[Accountancy](#)[Business Studies](#)[Economics](#)[Management](#)[Income Tax](#)[Finance](#)[Statistics for Economics](#)

SSC/ BANKING

[SSC CGL Syllabus](#)[SBI PO Syllabus](#)[SBI Clerk Syllabus](#)[IBPS PO Syllabus](#)[IBPS Clerk Syllabus](#)[Aptitude Questions](#)[SSC CGL Practice Papers](#)

GfG School

[CBSE Notes for Class 8](#)[CBSE Notes for Class 9](#)[CBSE Notes for Class 10](#)[CBSE Notes for Class 11](#)[CBSE Notes for Class 12](#)[English Grammar](#)

UPSC

[Polity Notes](#)[Geography Notes](#)[History Notes](#)[Science and Technology Notes](#)[Economics Notes](#)[Important Topics in Ethics](#)[UPSC Previous Year Papers](#)

Write & Earn

[Write an Article](#)[Improve an Article](#)[Pick Topics to Write](#)[Write Interview Experience](#)[Internships](#)

@geeksforgeeks , Some rights reserved