**Spring Web MVC**

-> It is one module in Spring Framework to develop web applications.

-> Web MVC module simplified web application development process.

1) Form Binding ( form <---> java obj )

2) Flexibility in Form Binding (type conversion)

3) Multiple Presentation Technologies (JSP & Thymeleaf)

4) Form Tag Library (ready-made tags support)

Note: To develop web application using spring-boot we need to add below starter in pom.xml

**### spring-boot-starter-web ###**

-> The above starter provides support for below things

1) MVC based web applications

2) RESTFul Services

3) Embedded Container (Tomcat)

**Spring Web MVC Architecture**

1) DispatcherServlet

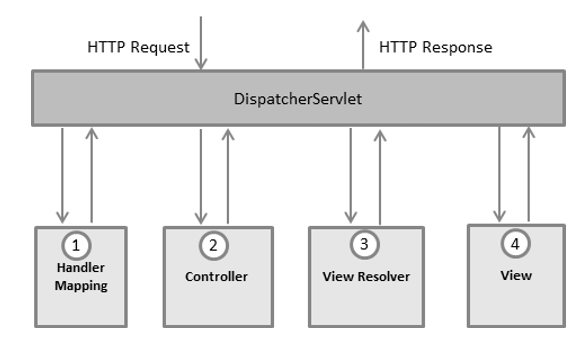
2) HandlerMapper

3) Controller

4) ModelAndView

5) ViewResolver

6) View



=> DispatcherServlet : Framework Servlet / Front Controller.

### Responsible to perform Pre-Processing and Post-Processing of request

=> Handler Mapper : Responsible to identify Request Handler class (controller)

=> Controller: Java class which is responsible to handle request & response

### Controller will return ModelAndView object to DispatcherServlet.

Model: Represents data in key-value format

View: Logical File Name

Note: Controllers are loosely coupled with Presentation technology.

ViewResolver: To identify presentation file location and technology

View: It is responsible to render Model data in view file.

**Building First Web App with Spring Boot**

1) Create Spring Starter Project with below dependencies

a) spring-boot-starter-web

b) spring-boot-devtools

c) tomcat-embed-jasper (mvnrepository.com)

2) Create controller class with required methods & map controller methods to URL pattern

3) Create View File with presentation logic

4) Configure View Resolver in application.properties file

5) Run the application and test it.

**Observations**

-> devtools dependency is used to restart our server when we make code changes.

-> To represent java class as controller we are using @Controller annotation

-> Controller methods we need to map with HTTP methods using unique URL pattern

GET --> @GetMapping

POST --> @PostMapping

-> Apache Tomcat is coming as default embedded container.

-> Embedded container port number is 8080. We can change that port number using application.properties file

server.port = 9090

-> Spring Boot web apps will not have context path. We can add context-path using application.properties file.

server.servlet.context-path=/ashokit

**Application Code**

