**Spring MVC 🡪**

* **Windows application**
* **Web application🡪** 
  + **Static web pages application (HTML)**
  + **Dynamic web application**

**Spring MVC is used to develop dynamic web applications. Spring MVC is derived from Spring therefore all core functionality of Spring core will be inherited so development will become very easy.**

**Web Application we require server.**

**What is server?**

**Request**

Server

Logic

Client

**response**

Data

Jsp -Java server pages

Servelet – is also a java class.

Spring MVC internally going to implement jsp servlet approach.

**MVC - > design pattern**

Model -> POJO which is going to get data from database

View -> presentation layer

Controller -> servlet – java class

Jsp Servlet Approach

Controller 1

Spring MVC approach

Spring MVC

1. Spring MVC we used developing web applications.
2. Spring MVC is built on top servlet
3. Jsp and servlet
4. Jsp – views
5. Servlet – controllers.
6. Spring MVC implementation

Controller 1

FrontController

(Servlet)

Client

Controller 2

(data and , view2 info)

Controller 3

View Resolver

View3

View

2

View1

index

indexController

getIndex()

index

name=”Bhushan”

View Resolver index

Front Controller

Client

Spring MVC - It is use for developing web application so we need server.

1. Front Controller / DispatcherServlet
2. View Resolver
3. Controller
4. Handlers - methods
5. Views

Core concepts of servlet

* HttpServlerRequest request
* HttpServletResponse response

View – jsp

In jsp pages we will be able to embed html and expression language to get data from controllers.

Implementation of Spring MVC application

There are some common steps we will have to follow.

(we will have to add dependencies in POM.xml file for spring MVC)

1. Configure Front Controller -> it is also called as Dispatcher Servlet. And we will have to configure this in web.xml
2. Configure view resolver in configuration xml file.

In view resolver we will have to properties

Prefix

suffix

1. Create package and add controllers

We will have to tell compiler that this is not a simple java class please treat this this class as a controller.

@Controller

1. Create views that is nothing but jsp pages in a folder which you can create in

Before you were getting data from controller and printing in console so same thing we will have to do here, we will have to print the data from controller to jsp pages

Data sending from controller to jsp and we will have to display this data in jsp view.

Sending data from controller to views.

In Spring MVC for achieving my requirement we have two ways

1. Model 🡪
   1. Method by the name

addAttribute(String key, Object value);

1. ModelAndView 🡪
   1. Method by the name

addObject(String key, Object value);

here we don’t have to return view name because here we wre going to return object of ModelAndView type. And we have a method setViewName(), using which we can set the name of view.

Scriptlet use to write javacode in jsp pages

<% %>

Expression language to get data from controller🡪 ${}

Second Req. I want to send data from view to controller.

* Forms are best and commonly use choice for sending data from view to controller

<form action=”/process” method=”POST”>

Name = <input type=”text” name=”uname”>

Email = <input type=”email” name=”email”>

Password = <input type=”password” name=”password”>

<button >submit</>button

</form>

When client is sending data from client side that view to controller at that time we can collect the data in controller using two ways

@RequestParam

@ModelAttribute

In core servlet jsp we can get data in controller using following method

request.getParameter("uname");

@RequestParam – using @RequestParam we will have to get all values one by one

@ModelAttribute – using @ModelAttribute

We can get all values at once

Important points in Spring MVC