**WebController.java**

**package** com.doj.spring.web.controller;

**import** java.util.HashMap;

**import** java.util.Map;

**import** javax.servlet.http.HttpServletRequest;

**import** org.springframework.stereotype.Controller;

**import** org.springframework.ui.ModelMap;

**import** org.springframework.web.bind.annotation.PathVariable;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestMethod;

**import** org.springframework.web.bind.annotation.RequestParam;

**import** org.springframework.web.servlet.ModelAndView;

**import** com.doj.spring.web.bean.Student;

@Controller

**public** **class** WebController {

//multiple mapping with one remote method

@RequestMapping(value={"/", "/index","/home","/welcome"})

**public** String home(){

**return** "home";

}

//With Model and Model Name to View Resolver

@RequestMapping("/indexc")

**public** ModelAndView welcome(){

Map<String, String> model = **new** HashMap<>();

model.put("name", "Sumit");

**return** **new** ModelAndView("home","model", model);

}

//We are using Spring ModelMap for return the model value

@RequestMapping("/hello")

**public** String index(ModelMap model){

model.put("name", "Sumit");

**return** "home";

}

//We are using Spring ModelMap and fetching request parameter here

@RequestMapping("/doj")

**public** String hello(ModelMap model, HttpServletRequest request){

String name = request.getParameter("name");

model.put("name", name);

**return** "home";

}

//We are using Spring ModelMap and Mapping the attribte with request param annotation with attributes whatever is ur requirement

@RequestMapping("/dojc")

**public** String doj(ModelMap model, @RequestParam(defaultValue = "DOJ Students", required=**true**, value="fname") String name,

@RequestParam(required = **false**, value="lname") String sname){

**if**(sname != **null**){

name = name +" "+ sname;

}

model.put("name", name);

**return** "home";

}

@RequestMapping("/doj-student-{fname}-{lname}")

**public** String dojStudent(ModelMap model, @PathVariable(value="fname") String name, @PathVariable(value="lname") String sname,

@RequestParam String address){

**if**(sname != **null**){

name = name +" "+ sname;

}

model.put("name", name);

**return** "home";

}

@RequestMapping(value="/doj-student", method=RequestMethod.*GET*)

**public** String getDojStudent(ModelMap model, Student student){

String name = **null**;

**if**(student.getFname() != **null**){

name = student.getFname();

}

**if**(student.getLname() != **null**){

name = name + " " +student.getLname();

}

**if**(student.getAddress() != **null**){

name = name+" "+student.getAddress();

}

**if**(student.getCourse() != **null**){

name = name+" "+student.getCourse();

}

model.put("name", name);

**return** "home";

}

@RequestMapping(value = "/dojstudent", method=RequestMethod.*GET*)

**public** String student(){

**return** "student";

}

@RequestMapping(value="/dojstudent", method=RequestMethod.*POST*)

**public** String getStudent(ModelMap model, Student student){

String name = **null**;

**if**(student.getFname() != **null**){

name = student.getFname();

}

**if**(student.getLname() != **null**){

name = name + " " +student.getLname();

}

**if**(student.getAddress() != **null**){

name = name+" "+student.getAddress();

}

**if**(student.getCourse() != **null**){

name = name+" "+student.getCourse();

}

model.put("name", name);

**return** "home";

}

}

**Student.java**

**package** com.doj.spring.web.bean;

**public** **class** Student {

String fname;

String lname;

String address;

String course;

**public** String getFname() {

**return** fname;

}

**public** **void** setFname(String fname) {

**this**.fname = fname;

}

**public** String getLname() {

**return** lname;

}

**public** **void** setLname(String lname) {

**this**.lname = lname;

}

**public** String getAddress() {

**return** address;

}

**public** **void** setAddress(String address) {

**this**.address = address;

}

**public** String getCourse() {

**return** course;

}

**public** **void** setCourse(String course) {

**this**.course = course;

}

}

**home.jsp**

<h1>${name} Welcome to DOJ Classes for Spring MVC!!!</h1>

**student.jsp**

<h1>Student Form</h1><br/>

<form action=*"dojstudent"* method=*"post"*>

First Name : <input name=*"fname"* type=*"text"*><br/>

Last Name : <input name=*"lname"* type=*"text"*><br/>

Address : <input name=*"address"* type=*"text"*><br/>

Course : <input name=*"course"* type=*"text"*><br/>

<input type=*"Submit"*>

</form>

**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"*>

<display-name>SpringWebApplication</display-name>

<servlet>

<servlet-name>springwebapp</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<!-- <init-param>

<param-name></param-name>

<param-value>/WEB-INF/xyz-servlet.xml</param-value>

</init-param> -->

</servlet>

<servlet-mapping>

<servlet-name>springwebapp</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

</web-app>

**springwebapp-servlet.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"*

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"*>

<!-- Component scaning from this base package -->

<context:component-scan base-package=*"com.doj.spring.web.controller"*/>

<context:annotation-config/>

<bean class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>

<property name=*"prefix"* value=*"/WEB-INF/view/"*></property>

<property name=*"suffix"* value=*".jsp"*></property>

</bean>

</beans>

Explanation:-

**Point1:-**

We were using simple web application at that time we use web.xml to map url with servlet with tag url-mapping.

Same thing we can here by

@RequestMapping("/hello")

**public** String home(){

**return** "home";

}

We will run application and try to access url **/hello** then **home** method would be called and it will return home and then DS look into **springwebapp-servlet.xml** and add .jsp at the end of home then call home.jsp file and show the result there.

Simply we are binding /hello url with home.jsp page.

@RequestMapping("/") this is default binding. When we access url with any parameter then it will show home.jsp page like (i.e. localhost:8080/AppName)

**Point2:-**

We can also bind multiple url with single method like below:

//multiple mapping with one remote method

@RequestMapping(value={"/", "/index","/home","/welcome"})

**public** String home(){

**return** "home";

}

**Point3:-**

ModelAndView is a class which is used to return model with the view name in Spring2.5 version and below this version we have to use ModelAndView to return model but int 3.2 version we can return also String as well.

Can pass Model name to get value from Model object without name also we can use model by default it add into model.

//With Model and Model Name to View Resolver

@RequestMapping("/indexc")

**public** ModelAndView welcome(){

Map<String, String> xyz = **new** HashMap<>();

model.put("name", "Sumit");

**return** **new** ModelAndView("home", xyz);

}

Here create simple Map and then creating ModelAndView with that map then that will we return. My model is xyz and view is home then I create object of ModelAndView with these values.

Here we are setting sumit value into name field and add into Model then we can access this in home.jsp pass. Like this

<h1>${name} Welcome to DOJ Classes for Spring MVC!!!</h1>

In above scenario we did not give model name so we were able to access name value without any model name automatically model present there. but we can also give model name then we have to access with that model name other can not be accessed.

//With Model and Model Name to View Resolver

@RequestMapping("/indexc")

**public** ModelAndView welcome(){

Map<String, String> model = **new** HashMap<>();

model.put("name", "Sumit");

**return** **new** ModelAndView("home","model", model);

}

Now by this can not access name field’s value

<h1>${name} Welcome to DOJ Classes for Spring MVC!!!</h1>

To access value now we have to use model name here

<h1>${model.name} Welcome to DOJ Classes for Spring MVC!!!</h1>

In above scenario we are creating Model Map then put value and return that ModelAndView. But we can also use spring Model Map. In this can no need to return ModelAndView it is automatically attached with view name. like below

//We are using Spring ModelMap for return the model value

@RequestMapping("/hello")

**public** String index(ModelMap model){

model.put("name", "Sumit");

**return** "home";

}

**Point4:-**

Now We want to pass name with request ulr and use that in view.

If we want to pass request parameter and we want to access that then we use

//We are using Spring ModelMap and fetching request parameter here

@RequestMapping("/doj")

**public** String hello(ModelMap model, HttpServletRequest request){

String name = request.getParameter("name");

model.put("name", name);

**return** "home";

}

This is like simple http servlet pass http request and get parameter from that.

We access url “localhost:8080/Appname/doj” then No name would be displayed but if we access “localhost:8080/Appname/doj?name=sumit” then name would be displayed.

Instead of using HttpServletRequest we want to use @RequestParam same it will work also.

@RequestMapping("/dojc")

**public** String doj(ModelMap model, @RequestParam String name){

model.put("name", name);

**return** "home";

}

name should be same as in method parameter other wise will not work like we you pass

“localhost:8080/Appname/doj?name1=sumit” will not displayed.

**Point5:-**

If we use two request parameter then we have to also pass two parameter other wise it will show error like below:

@RequestMapping("/dojc")

**public** String doj(ModelMap model, @RequestParam String name, String sname){

**if**(sname != **null**){

name = name +" "+ sname;

}

model.put("name", name);

**return** "home";

}

If pass one parameter then will not run but we also have attribute to run application if we do not pass parameter just set field is required or not like below:

If you want to use mandatory parameter then use this

//We are using Spring ModelMap and Mapping the attribte with request param annotation with attributes whatever is ur requirement

@RequestMapping("/dojc")

**public** String doj(ModelMap model, @RequestParam(required=**true**) String name,

@RequestParam(required = **false**) String sname){

**if**(sname != **null**){

name = name +" "+ sname;

}

model.put("name", name);

**return** "home";

}

Default value Required is true.

If we are not passing any value then default message would be print.

//We are using Spring ModelMap and Mapping the attribte with request param annotation with attributes whatever is ur requirement

@RequestMapping("/dojc")

**public** String doj(ModelMap model, @RequestParam(defaultValue = "DOJ Students") String name, @RequestParam(required = **false**) String sname){

**if**(sname != **null**){

name = name +" "+ sname;

}

model.put("name", name);

**return** "home";

}

@RequestParam also have another attribute “value” that is use for security purpose we do not want to show are field name to user at that time we use value attribute. Whatever value we provide into “value” attribute by that only can access url that value is bind with method parameter like below:

//We are using Spring ModelMap and Mapping the attribte with request param annotation with attributes whatever is ur requirement

@RequestMapping("/dojc")

**public** String doj(ModelMap model, @RequestParam(defaultValue = "DOJ Students", required=**true**, value="fname") String name,

@RequestParam(required = **false**, value="lname") String sname)

{

**if**(sname != **null**){

name = name +" "+ sname;

}

model.put("name", name);

**return** "home";

}

If we use this url “localhost:8080/Appname/doj?fname=sumit” or “localhost:8080/Appname/doj?fname=sumit&&lname=saini” then it would be accessible otherwise not.

Fname value bind with name and lname value with sname.

**Point6:-**

We can also define path pattern by using @PathVariable. Define own url pattern

@RequestMapping("/doj-student-{name}-{name}")

**public** String dojStudent(ModelMap model, @PathVariable String name, @PathVariable String sname){

**if**(sname != **null**){

name = name +" "+ sname;

}

model.put("name", name);

**return** "home";

}

@RequestMapping("/doj-student-{fname}-{lname}")

**public** String dojStudent(ModelMap model, @PathVariable(value="fname") String name, @PathVariable(value="lname") String sname){

**if**(sname != **null**){

name = name +" "+ sname;

}

model.put("name", name);

**return** "home";

}

**Point7**:-

We use @RequestParam where we use more request parameter. And we used @PathVariable annotation for user friendly we do not use this in that condition where we use more request parameter. Both are different thing these are not alternate of each other and also can use both together.

But

public String hello(ModelMap model, HttpServletRequest request){

String name = request.getParameter("name");

It has alternative of @RequestParam.

@RequestMapping("/dojc")

**public** String doj(ModelMap model, @RequestParam(required=**true**) String name,

@RequestParam(required = **false**) String sname){

**if**(sname != **null**){

name = name +" "+ sname;

}

model.put("name", name);

**return** "home";

}

In this case name field is mandatory with the request. If we will not pass then it will through exception. But sir name field is not mandatory because it not required.

But user may be uneducated if he will get error that is not good thing we are developer we have to handle also this scenario so to handle this we pass default value so that it not show Exception instead of exception it will show that message.

//We are using Spring ModelMap and Mapping the attribte with request param annotation with attributes whatever is ur requirement

@RequestMapping("/dojc")

**public** String doj(ModelMap model, @RequestParam(defaultValue = "DOJ Students", required=**true**, value="fname") String name,

@RequestParam(required = **false**, value="lname") String sname){

**if**(sname != **null**){

name = name +" "+ sname;

}

model.put("name", name);

**return** "home";

}

@RequestMapping("/doj-student-{fname}-{lname}")

**public** String dojStudent(ModelMap model, @PathVariable(value="fname") String name, @PathVariable(value="lname") String sname,

@RequestParam String address){

**if**(sname != **null**){

name = name +" "+ sname;

}

model.put("name", name);

**return** "home";

}

**Point8**:-

If we have more request parameter in that case we should not use url pattern or @RequestParam annotation instead of this we should use bean.

In Spring2.5 below version we used parameter URL or Command Class but above 2.5 we use bean.

We should use for long request parameter but one thing must remember request parameter field should match with bean properties. Because it automatically bind otherwise it will not work. Bean is not a string bean is Pojo Class.

@RequestMapping(value="/doj-student", method=RequestMethod.*GET*)

**public** String getDojStudent(ModelMap model, Student student){

String name = **null**;

**if**(student.getFname() != **null**){

name = student.getFname();

}

**if**(student.getLname() != **null**){

name = name + " " +student.getLname();

}

**if**(student.getAddress() != **null**){

name = name+" "+student.getAddress();

}

**if**(student.getCourse() != **null**){

name = name+" "+student.getCourse();

}

model.put("name", name);

**return** "home";

}

@RequestMapping(value = "/dojstudent", method=RequestMethod.*GET*)

**public** String student(){

**return** "student";

}

To access above get method we pass parameter with in url like below:

“localhost:8080/Appname/doj?fname=sumit&lname=saini&address=Delhi&course=MCA”

@RequestMapping(value="/dojstudent", method=RequestMethod.*POST*)

**public** String getStudent(ModelMap model, Student student){

String name = **null**;

**if**(student.getFname() != **null**){

name = student.getFname();

}

**if**(student.getLname() != **null**){

name = name + " " +student.getLname();

}

**if**(student.getAddress() != **null**){

name = name+" "+student.getAddress();

}

**if**(student.getCourse() != **null**){

name = name+" "+student.getCourse();

}

model.put("name", name);

**return** "home";

}

}

Above method url is worked with post method when we submit form

**student.jsp**

<h1>Student Form</h1><br/>

<form action=*"dojstudent"* method=*"post"*>

First Name : <input name=*"fname"* type=*"text"*><br/>

Last Name : <input name=*"lname"* type=*"text"*><br/>

Address : <input name=*"address"* type=*"text"*><br/>

Course : <input name=*"course"* type=*"text"*><br/>

<input type=*"Submit"*>

</form>