SpringMVC传递参数的方式：

1. servlet API(引入servlet-api.jar)

Public ModelAndView index(HttpServletRequest request,HttpSession session){

ModelAndView mv = new ModelAndView ();

String name = request.getParameter(“name”);

session.setAttribute(“name”,”value”);

mv.setViewName(“index”);

return mv;

}

Jsp:取值

<%=request.getSession().getAttribute(“name”)%>

2、@RequestParam注解

Public ModelAndView index(@RequestParam(“student\_name”) StringstudentName){

ModelAndView mv = new ModelAndView ();

System.out.print(name);

mv.setViewName(“index”);

return mv;

}

1. 设置Session的值

@Controller

@SessionAttributes(“name”)

Public class MyController{

Public ModelAndView index(ModelMap model){

ModelAndView mv = new ModelAndView ();

model.addAttribute(“name”,”value”);

//mv.addObject(“name”,”value”);

return mv;

}

}

Jsp:取值

<%=request.getSession().getAttribute(“name”)%>

\*\*\*\*

这里我们仅将一个 ModelMap 的属性放入 Session 中，其实 @SessionAttributes 允许指定多个属性。

1. 通过字符串数组的方式指定多个属性：

@SessionAttributes({“attr1”,”attr2”})

1. 通过属性类型指定要session化的ModelMap属性：

@SessionAttributes(types = User.class)

1. 指定多个类：

@SessionAttributes(types = {User.class,Dept.class})

4、联合使用属性名和属性类型指定：

@SessionAttributes(types{User.class,Dept.class},

value={“attr1”,”attr2”}

1. 映射POJO传值

Public ModelAndView index(Student student){

ModelAndView mv = new ModelAndView ();

String name = student.getName();

String age = student.getAge();

mv.setViewName(“student”);

return mv;

}

1. 使用URL传递参数

@RequestMapping(“/my/{studentId}”)

Public ModelAndView index(@PathVariable(“studentId”) Integer studentId){

ModelAndView mv = new ModelAndView ();

System.out.print(studentId);

mv.setViewName(“student”);

return mv;

}