

**SpringMVC框架把原来Servlet中的请求转发和重定向操作进行了封装，使用更简单的方式实现请求转发和重定向**

**用一个关键字forward：表示转发，实现**  
**request.getRequestDispatcher("xx.jsp").forward(request, response);**

**用关键字redirect：表示重定向，实现**  
**response.sendRedirect("xx.jsp");**

请求的转发：

一个请求请求到资源1，资源1在转发给资源2，资源再处理请求，请求次数是1次，所以使用的都是同一个HttpServletRequest和HttpServletResponse对象，是在项目内部进行的，所以可以访问项目的WEB-INF目录中的内容，即资源2可以是WEB-INF中的资源

请求的重定向：

一个请求像资源1发起请求，资源1给浏览器一个资源2的地址，浏览器再发送一个请求访问资源2，这是2次请求。两次请求都相当于是用户直接输入URL地址发起的，所以都不能访问WEB-INF中的资源。第二次的资源请求是无法获取第一次请求中获取的存储到request作用域中的数据。

**很明显，关键字forward和redirect不和视图解析器一同使用**

## 使用方式

**forward:**

当我们的处理器方法返回ModelAndView时，实现forward转发

语法: **setViewName("forward:视图文件完整路径")**

**setViewName("forward:WEB-INF/static/jsp/xx.jsp");**

特点: 不和视图解析器一同使用，即使配置了，也当作没有

作用: 当由于业务需求，我们的视图资源不能放到视图解析器中配置的拼接路径时，使用这种方式完成转发

**redirect:**

当我们的处理器方法返回ModelAndView时，实现redirect重定向

语法: **setViewName("redirect:视图文件完整路径")**

**setViewName("redirect:static/jsp/xx.jsp");**

特点: 不和视图解析器一同使用，即使配置了，也当作没有

SpringMVC框架对重定向操作的特点:

1、框架会把ModelAndView中的Model中存储的简单类型数据，转为简单的String类型使用，然后作为重定向的请求地址的参数使用，目的是让两次的请求之间传递数据。如我们访问地址:

**http://localhost:8080/d\_SSM\_04/** 该地址进入首页，然后输入参数点击提交到资源1地址

**http://localhost:8080/d\_SSM\_04/some.do?name=zhangsan&age=4** 该地址中重定向到一个jsp页面地址，附带参数 (前提是这个请求中的参数被从该请求的request对象中获取，)，如下

**http://localhost:8080/d\_SSM\_04/static/jsp/show.jsp?name=zhangsan&age=4**

当我们这个jsp页面想要获取附带的参数时，页面中的EL表示为**`${param.name}`**，

底层是**`<%=request.getParameter("name")%>`**

`redirect`不能访问WEB-INF目录下受保护的资源