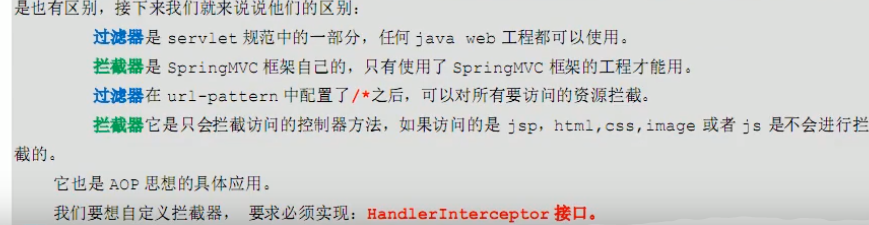
1. 拦截器和过滤器的区别：

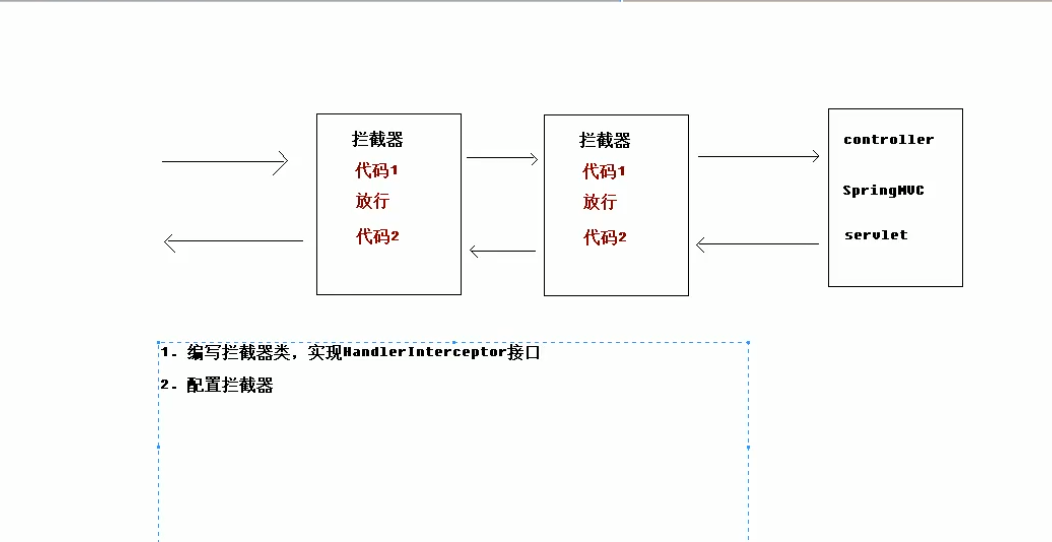


即：

过滤器配置了/\*什么资源(jsp、js、controller)都能拦截，是servlet的技术

拦截器只能拦截controller，是SpringMVC中的技术。

拦截器 ⊆ 过滤器



1. 拦截器，首先编写demo的jsp页面

|  |
| --- |
| <**a href="user/testInterceptor"**>拦截器测试。。</**a**> |

1. 自定义拦截器，需要实现HandlerInterceptor类，并重写里面的

preHandle、postHandle、afterCompletion三个方法。。分别是controller之前，之后且响应之前，以及页面响应后执行。

如果需要两个拦截器，就编写两个自定义拦截器MyInterCeptor01、MyInterCeptor02

|  |
| --- |
| **package** com.igeek.shop.interceptor;  **import** org.springframework.web.servlet.HandlerInterceptor; **import** org.springframework.web.servlet.ModelAndView;  **import** javax.servlet.http.HttpServletRequest; **import** javax.servlet.http.HttpServletResponse;  **public class** MyInterCeptor01 **implements** HandlerInterceptor {  */\*  \* 预处理。执行controller之前，return true即放行  \* \*/* @Override  **public boolean** preHandle(HttpServletRequest httpServletRequest, HttpServletResponse httpServletResponse, Object o) **throws** Exception {  System.***out***.println(**"执行了MyInterCeptor01.preHandle().........."**);  *//return true;   //模拟false  //httpServletRequest.getRequestDispatcher("/WEB-INF/pages/error.jsp").forward(httpServletRequest,httpServletResponse);* **return true**;   }   */\*  \* 在controller执行后，再次拦截。。。 响应jsp执行之前。。。。。  \* 如果这里也有jsp跳转，将不跳转controller对应的success.jsp  \* \*/* @Override  **public void** postHandle(HttpServletRequest httpServletRequest, HttpServletResponse httpServletResponse, Object o, ModelAndView modelAndView) **throws** Exception {  System.***out***.println(**"执行了MyInterCeptor01.postHandle().........."**);  *//httpServletRequest.getRequestDispatcher("/WEB-INF/pages/error.jsp").forward(httpServletRequest,httpServletResponse);* }   */\*  \*在页面相应之后执行，可以用来关闭连接。  \* \*/* @Override  **public void** afterCompletion(HttpServletRequest httpServletRequest, HttpServletResponse httpServletResponse, Object o, Exception e) **throws** Exception {  System.***out***.println(**"执行了MyInterCeptor01.afterCompletion().........."**);   } } |
| **package** com.igeek.shop.interceptor;  **import** org.springframework.web.servlet.HandlerInterceptor; **import** org.springframework.web.servlet.ModelAndView;  **import** javax.servlet.http.HttpServletRequest; **import** javax.servlet.http.HttpServletResponse;  **public class** MyInterCeptor02 **implements** HandlerInterceptor {  */\*  \* 预处理。执行controller之前，return true即放行  \* \*/* @Override  **public boolean** preHandle(HttpServletRequest httpServletRequest, HttpServletResponse httpServletResponse, Object o) **throws** Exception {  System.***out***.println(**"执行了MyInterCeptor02.preHandle().........."**);  *//return true;   //模拟false  //httpServletRequest.getRequestDispatcher("/WEB-INF/pages/error.jsp").forward(httpServletRequest,httpServletResponse);* **return true**;   }   */\*  \* 在controller执行后，再次拦截。。。 响应jsp执行之前。。。。。  \* 如果这里也有jsp跳转，将不跳转controller对应的success.jsp  \* \*/* @Override  **public void** postHandle(HttpServletRequest httpServletRequest, HttpServletResponse httpServletResponse, Object o, ModelAndView modelAndView) **throws** Exception {  System.***out***.println(**"执行了MyInterCeptor02.postHandle().........."**);  *//httpServletRequest.getRequestDispatcher("/WEB-INF/pages/error.jsp").forward(httpServletRequest,httpServletResponse);* }   */\*  \*在页面相应之后执行，可以用来关闭连接。  \* \*/* @Override  **public void** afterCompletion(HttpServletRequest httpServletRequest, HttpServletResponse httpServletResponse, Object o, Exception e) **throws** Exception {  System.***out***.println(**"执行了MyInterCeptor02.afterCompletion().........."**);   } } |

1. 在springMVC-config.xml配置拦截器。

暂时配置了两个。

<**mvc:mapping path="/user/\*\*/"**/> 表示拦截user之下的资源。。

|  |
| --- |
| <**mvc:interceptors**>  <**mvc:interceptor**>  *<!--要拦截的方法-->* <**mvc:mapping path="/user/\*\*/"**/> *<!-- /\*\*表示拦截所有的controller -->   <!--不拦截的方法 --> <!-- user/\*\*表示拦截user下的controller -->  <!--<mvc:exclude-mapping path=""/>-->   <!--配置拦截器-->* <**bean id="myInterCeptor01" class="com.igeek.shop.interceptor.MyInterCeptor01"**/>  </**mvc:interceptor**>   *<!--配置第二个拦截器-->* <**mvc:interceptor**>  *<!--要拦截的方法-->* <**mvc:mapping path="/\*\*/"**/> *<!-- /\*\*表示拦截所有的controller -->  <!--配置拦截器-->* <**bean id="myInterCeptor02" class="com.igeek.shop.interceptor.MyInterCeptor02"**/>  </**mvc:interceptor**> </**mvc:interceptors**> |

1. controller编写

|  |
| --- |
| */\*  \* 拦截器测试  \* \*/* @RequestMapping(path = **"/testInterceptor"**) **public** String testInterceptor() {  System.***out***.println(**"进入FileUploadController.testInterceptor()方法了。。。。。"**);    **return "success"**; } |