**package** com.mvc.interceptors;  
  
**import** org.springframework.lang.Nullable;  
**import** org.springframework.stereotype.Component;  
**import** org.springframework.web.servlet.HandlerInterceptor;  
**import** org.springframework.web.servlet.ModelAndView;  
  
**import** javax.servlet.http.HttpServletRequest;  
**import** javax.servlet.http.HttpServletResponse;  
  
*/\*\*  
 \** ***@Description:*** *拦截器拦截用户有没有登录  
 \** ***@author:*** *LuShao  
 \** ***@create:*** *2018-04-11 16:58  
 \*\*/  
//@Component("loginInterceptor")***public class** LoginInterceptor **implements** HandlerInterceptor {  
 @Override  
 **public boolean** preHandle(HttpServletRequest request, HttpServletResponse response, Object handler) **throws** Exception {  
 String requestURI = request.getRequestURI();  
 System.***out***.println(**"拦截器Interceptor:"**+requestURI);  
 **return true**;  
 }  
  
 @Override  
 **public void** postHandle(HttpServletRequest request, HttpServletResponse response, Object handler, @Nullable ModelAndView modelAndView) **throws** Exception {  
  
 }  
  
 @Override  
 **public void** afterCompletion(HttpServletRequest request, HttpServletResponse response, Object handler, @Nullable Exception ex) **throws** Exception {  
  
 }  
}

**web.xml**

*<!--配置springMVC 的 DispatcherServlet-->*<**servlet**>  
 <**servlet-name**>springMVC</**servlet-name**>  
 <**servlet-class**>org.springframework.web.servlet.DispatcherServlet</**servlet-class**>  
 *<!--加载配置文件-->* <**init-param**>  
 <**param-name**>contextConfigLocation</**param-name**>  
 <**param-value**>classpath:conf/spring-mvc.xml</**param-value**>  
 </**init-param**>  
 <**load-on-startup**>1</**load-on-startup**>  
 *<!--<async-supported>true</async-supported>-->*</**servlet**>  
<**servlet-mapping**>  
 <**servlet-name**>springMVC</**servlet-name**>  
 <**url-pattern**>\*.do</**url-pattern**>  
</**servlet-mapping**>

**这里的配置的.do**

**所以interceptor只是拦截这个后缀的地址**

*<!--配置Interceptor-->*<**mvc:interceptors**>  
 <**mvc:interceptor**>  
 <**mvc:mapping path="/\*\*"**/>  
 <**bean class="com.mvc.interceptors.TokenInterceptor"** />  
 </**mvc:interceptor**>  
</**mvc:interceptors**>  
<**mvc:interceptors**>  
 <**mvc:interceptor**>  
 <**mvc:mapping path="/\*\*"**/>  
 <**bean class="com.mvc.interceptors.LoginInterceptor"** />  
 </**mvc:interceptor**>  
</**mvc:interceptors**>

**注意：这里有先后顺序的**

**过滤器Filter**

**package** com.mvc.filters;  
  
**import** org.springframework.web.filter.OncePerRequestFilter;  
  
**import** javax.servlet.FilterChain;  
**import** javax.servlet.ServletException;  
**import** javax.servlet.http.HttpServletRequest;  
**import** javax.servlet.http.HttpServletResponse;  
**import** java.io.IOException;  
  
*/\*\*  
 \** ***@Description:*** *过滤特定jsp页面  
 \** ***@author:*** *LuShao  
 \** ***@create:*** *2018-04-11 16:53  
 \*\*/***public class** JspFilter **extends** OncePerRequestFilter {  
 @Override  
 **protected void** doFilterInternal(HttpServletRequest httpServletRequest, HttpServletResponse httpServletResponse, FilterChain filterChain) **throws** ServletException, IOException {  
 String uri=httpServletRequest.getRequestURI();  
 System.***out***.println(**"JspFilter doFilterInternal过滤器："**+uri);  
*// if(uri.contains(".jsp")){  
// if(uri.contains("login")){  
// filterChain.doFilter(httpServletRequest,httpServletResponse);  
// }  
// }* filterChain.doFilter(httpServletRequest,httpServletResponse);  
 }  
}

**过滤器要在web.xml里面配置**

*<!--配置编码格式-->*<**filter**>  
 <**filter-name**>encodeFilter</**filter-name**>  
 <**filter-class**>org.springframework.web.filter.CharacterEncodingFilter</**filter-class**>  
 <**init-param**>  
 <**param-name**>encoding</**param-name**>  
 <**param-value**>utf-8</**param-value**>  
 </**init-param**>  
</**filter**>  
<**filter-mapping**>  
 <**filter-name**>encodeFilter</**filter-name**>  
 <**url-pattern**>/\*</**url-pattern**>  
</**filter-mapping**>  
  
*<!--配置spring过滤器-->*<**filter**>  
 <**filter-name**>jspFilter</**filter-name**>  
 <**filter-class**>com.mvc.filters.JspFilter</**filter-class**>  
</**filter**>  
<**filter-mapping**>  
 <**filter-name**>jspFilter</**filter-name**>  
 *<!--<url-pattern>/\*</url-pattern>-->* <**url-pattern**>\*.jsp</**url-pattern**>  
</**filter-mapping**>

**拦截静态文件**

**Web.xml**

*<!--配置springMVC 的 DispatcherServlet-->*<**servlet**>  
 <**servlet-name**>springMVC</**servlet-name**>  
 <**servlet-class**>org.springframework.web.servlet.DispatcherServlet</**servlet-class**>  
 *<!--加载配置文件-->* <**init-param**>  
 <**param-name**>contextConfigLocation</**param-name**>  
 <**param-value**>classpath:conf/spring-mvc.xml</**param-value**>  
 </**init-param**>  
 <**load-on-startup**>1</**load-on-startup**> 这里的1表示 如果是正数则加载的时候就先实例化 如果是负数则运行的时候实例化  
 *<!--<async-supported>true</async-supported>-->*</**servlet**>  
<**servlet-mapping**>  
 <**servlet-name**>springMVC</**servlet-name**>  
 <**url-pattern**>\*.do</**url-pattern**>  
</**servlet-mapping**>

如果这里的是指定 非静态文件后缀 则不需要在spring配置文件中配置过滤静态文件的配置

**反之**<**url-pattern**>/</**url-pattern**>  
**下面是两种方式选择其中一种方式在spring配置文件中配置**

*<!--  
 静态资源映射  
 把项目的静态资源放到statics文件夹下  
-->  
<!--<mvc:resources mapping="/js/\*\*" location="/statics/js/" />-->  
<!--<mvc:default-servlet-handler></mvc:default-servlet-handler>-->*

*<!--  
 对模型视图名称的解析，即为模型视图名称添加前后缀，文件夹最后一个斜杠（/）不能少 这里使用jsp  
 默认的视图解析器 在上边的解析错误时使用（默认使用html）  
 把jsp放到web-info下面为了安全性 服务器内部路径 外部无法直接访问  
-->*<**bean id="viewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver"**>  
 <**property name="viewClass" value="org.springframework.web.servlet.view.JstlView"** />  
 <**property name="prefix" value="/WEB-INF/view/"** />  
 <**property name="suffix" value=".jsp"** />  
</**bean**>