拦截器

1. 自定义拦截器
2. 写一个拦截器

写一个类，实现HandlerInterceptor接口即可，接口有三个抽象方法

**public** **class** HelloInterceptors **implements** HandlerInterceptor {

**该方法在渲染视图之后被调用，可以用于释放资源**

@Override

**public** **void** afterCompletion(HttpServletRequest request, HttpServletResponse response, Object obj, Exception e)

**throws** Exception {

System.***out***.println("[HelloInterceptors] afterCompletion");

}

**目标方法被调用之后，渲染视图之前被调用。**

**可以对请求域中的属性或者视图进行修改**

@Override

**public** **void** postHandle(HttpServletRequest request, HttpServletResponse response, Object obj,

ModelAndView modelAndView) **throws** Exception {

System.***out***.println("[HelloInterceptors] postHandle");

}

**在调用目标方法之前被调用，如果拦截器中有任意一个的该方法的返回值为false就直接结束请求了。可以做权限，日志，事务等**

@Override

**public** **boolean** preHandle(HttpServletRequest request, HttpServletResponse response, Object obj) **throws** Exception {

System.***out***.println("[HelloInterceptors] preHandle");

**return** **true**;

}

}

1. 在SpringMVC配置文件中配置该拦截器  
   <mvc:interceptors>

<bean class=*"crud.dao.interceptors.*

*HelloInterceptors"*></bean>

</mvc:interceptors>

所有拦截器都要在<mvc**:**interceptors>中配置

1. 拦截器配置
2. 配置拦截器作用的路径

<!-- **配置拦截器，所有拦截器都应该在这里面配置** -->

<mvc:interceptors>

<!-- **配置指定路径的拦截器** -->

<mvc:interceptor>

<!—**配置拦截器拦截的URL**-->

<mvc:mapping path=*"/list"*/>

**<!--不需要拦截的URL <mvc:exclude-mapping path="/list"/> -->**

<!-- **bean标签必须在后边，否则会出错** -->

<bean class=*"crud.dao.interceptors.HelloInterceptors"*></bean>

</mvc:interceptor>

</mvc:interceptors>

1. 多个拦截器执行顺序

* preHandle方法按配置顺序的顺序执行
* postHandle方法按配置顺序的反序执行
* afterCompletion方法按配置顺序的反序执行

当有一个拦截器的preHandle方法返回false，它之后的拦截器都不会被执行，目标方法也不会被执行，但是在它之前执行过的拦截器的afterCompletion方法仍然会被执行