# **SpringMVC中的拦截器介绍**

1、SpringMvc中的拦截器：

SpringMvc拦截器帮我们按照一定规则拦截请求，后根据开发人员自定义的拦截逻辑进行处理；

自定义拦截器需要实现HandlerInterceptor接口；

自定义的拦截器实现类需要在SpringMvc配置文件中配置；

可以配置多个拦截器，配置的顺序会影响到拦截器的执行顺序，配置在前的先执行；

HandlerInterceptor有3个方法：

preHandle 预处理：在拦截方法前执行；

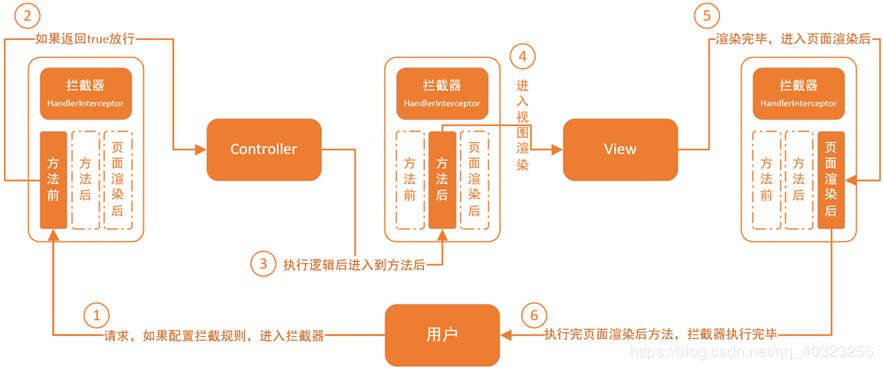
postHandle 后处理：在拦截方法后执行；

afterCompletion 渲染后处理：在页面渲染后执行；

拦截器也体现了AOP思想；

拦截器的应用：权限检查，日志记录，性能检测等；

1. 拦截器的执行流程图：

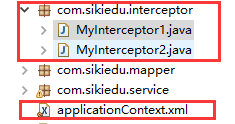


1. 总结的几条拦截器规则：
2. preHandle 预处理：根据拦截器定义的顺序，正向执行

可以考虑做权限，日记，事务，登陆等

2.postHandle 后处理：根据拦截器定义的顺序，逆向执行。需要所有的preHandle都返回true时才会调用。可以对请求域中的属性或视图做出修改

3.afterCompletion 渲染后处理：根据拦截器定义的顺序，逆向执行。preHandle返回true就会调用



MyInterceptor1.java:

|  |
| --- |
| public class MyInterceptor1 implements HandlerInterceptor {    @Override  public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler)  throws Exception {  System.out.println("1 PreHandle预处理");  return true;  }  @Override  public void postHandle(HttpServletRequest request, HttpServletResponse response, Object handler,  ModelAndView modelAndView) throws Exception {  System.out.println("1 postHandle后处理");  }  @Override  public void afterCompletion(HttpServletRequest request, HttpServletResponse response, Object handler, Exception ex)  throws Exception {  System.out.println("1 afterCompletion页面渲染后处理");  }  } |

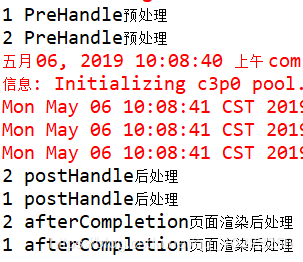
MyInterceptor2.java:

|  |
| --- |
| public class MyInterceptor2 implements HandlerInterceptor {  @Override  public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler)  throws Exception {  System.out.println("2 PreHandle预处理");  return true;  }  @Override  public void postHandle(HttpServletRequest request, HttpServletResponse response, Object handler,  ModelAndView modelAndView) throws Exception {  System.out.println("2 postHandle后处理");  }    @Override  public void afterCompletion(HttpServletRequest request, HttpServletResponse response, Object handler, Exception ex)  throws Exception {  System.out.println("2 afterCompletion页面渲染后处理");  }  } |

applicationContext.xml:

|  |
| --- |
| <mvc:interceptors>  <!-- 拦截器1 -->  <mvc:interceptor>  <mvc:mapping path="/\*\*"/>  <bean class="com.sikiedu.interceptor.MyInterceptor1"></bean>  </mvc:interceptor>    <!-- 拦截器2 -->  <mvc:interceptor>  <mvc:mapping path="/\*\*"/>  <bean class="com.sikiedu.interceptor.MyInterceptor2"></bean>  </mvc:interceptor>  </mvc:interceptors> |

****preHandle1返回true，preHandle2返回true：运行结果如下：****



****preHandle1返回true，preHandle2返回false：运行结果如下：****

