Http请求前后监听工具——HttpInterceptor

对请求的API在开始和结束进行处理。

注意：

1. 继承HandlerInterceptorAdaptor
2. 重写父类的三个方法（预处理、后处理、返回处理）

preHandle(HttpServletRequest request, HttpServletResponse response, Object handler)

postHandle(HttpServletRequest request, HttpServletResponse response, Object handler, ModelAndView modelAndView)

afterCompletion(HttpServletRequest request, HttpServletResponse response, Object handler, Exception ex)

*/\*\*  
 \* Http请求前后监听类  
 \* 对请求的API在开始和结束进行处理  
 \*/*@Slf4j  
**public class** HttpInterceptor **extends** HandlerInterceptorAdapter {  
  
 **private static final** String ***START\_TIME*** = **"requestStartTime"**;  
  
 */\*\*  
 \* 预处理（请求前）  
 \*  
 \** ***@param:*** *[request, response, handler]  
 \** ***@return:*** *boolean  
 \** ***@author:*** *nino  
 \** ***@date:*** *2019/5/27 16:27  
 \*/* @Override  
 **public boolean** preHandle(HttpServletRequest request, HttpServletResponse response, Object handler) **throws** Exception {  
 String url = request.getRequestURI();  
 *//获取请求参数对应的map* Map parameterMap = request.getParameterMap();  
 ***log***.info(**"request start. url:{}, params:{}"**, url, JsonMapper.*obj2String*(parameterMap));  
 **long** start = System.*currentTimeMillis*();  
 *//计算请求开始到结束消耗的时间* request.setAttribute(***START\_TIME***, start);  
 **return true**;  
 }  
  
 */\*\*  
 \* 后处理（正常请求会调用）  
 \* 场景：  
 \* 输出每次的请求信息、监控某个方法耗费事件  
 \*  
 \** ***@param:*** *[request, response, handler, modelAndView]  
 \** ***@return:*** *void  
 \** ***@author:*** *nino  
 \** ***@date:*** *2019/5/27 16:27  
 \*/* @Override  
 **public void** postHandle(HttpServletRequest request, HttpServletResponse response, Object handler, ModelAndView modelAndView) **throws** Exception {  
*// String url = request.getRequestURI();  
// long start = (Long) request.getAttribute(START\_TIME);  
// long end = System.currentTimeMillis();  
// log.info("request finished. url:{}, cost:{}", url, end - start);  
// removeThreadLocalInfo();* }  
  
 */\*\*  
 \* 返回处理（正常和异常都会调用）  
 \*  
 \** ***@param:*** *[request, response, handler, ex]  
 \** ***@return:*** *void  
 \** ***@author:*** *nino  
 \** ***@date:*** *2019/5/27 16:28  
 \*/* @Override  
 **public void** afterCompletion(HttpServletRequest request, HttpServletResponse response, Object handler, Exception ex) **throws** Exception {  
 String url = request.getRequestURI();  
 **long** start = (Long) request.getAttribute(***START\_TIME***);  
 **long** end = System.*currentTimeMillis*();  
 ***log***.info(**"request completed. url:{}, cost:{}"**, url, end - start);  
  
 removeThreadLocalInfo();  
 }  
  
 **public void** removeThreadLocalInfo() {  
 RequestHolder.*remove*();  
 }  
}

将此类注册到容器中：

