|  |
| --- |
| /\*\* |
|  | \* Description 公共区 操作日志 实现层 |
|  | \* Created by HealerJean |
|  | \* DATE 2017/5/19 15:11. |
|  | \*/ |
|  | @Transactional |
|  | @Service |
|  | public class OperateLogServiceImpl implements OperateLogService { |
|  |  |
|  | @Autowired |
|  | private OperateLogRespository operateLogRespository; |
|  |  |
|  | Logger logger = LoggerFactory.getLogger(OperateLogServiceImpl.class); |
|  | /\*\* |
|  | \* @Description 操作日志保存 |
|  | \* @Author HealerJean |
|  | \* @CreateDate 2017/5/19 15:26 |
|  | \*/ |
|  | @Override |
|  | public void addOptLog(OperateLog operateLog,String operateString) { |
|  | UserPrincipal userPrincipal = (UserPrincipal) SecurityUtils.getSubject().getPrincipal(); |
|  | operateLog.setOptId(userPrincipal.getId()); |
|  | operateLog.setOptName(userPrincipal.getRealName()); |
|  | HttpServletRequest request = ((ServletRequestAttributes) RequestContextHolder.getRequestAttributes()).getRequest(); |
|  |  |
|  |  |
|  | operateLog.setIp(getClientIp(request)); |
|  | //添加操作描述 例如 小明对ba9899d5-4d6d-47ee-96e8-aeaa44640631 进行了违约审核 |
|  | operateLog.setDescription(userPrincipal.getRealName()+"进行了"+ operateString+"的操作"); |
|  | operateLogRespository.save(operateLog); |
|  | } |
|  |  |
|  | /\*\* |
|  | \* @Description 日志中记录 操作者的ip |
|  | \* @Author HealerJean |
|  | \* @CreateDate 2017/5/19 16:46 |
|  | \*/ |
|  | public String getClientIp(HttpServletRequest request) { |
|  |  |
|  | String ip = request.getHeader("x-forwarded-for"); |
|  | //String ip = request.getHeader("X-real-ip"); |
|  | logger.info("x-forwarded-for = {}", ip); |
|  | if(ip == null || ip.length() == 0 || "unknown".equalsIgnoreCase(ip)) { |
|  | ip = request.getHeader("Proxy-Client-IP"); |
|  | logger.info("Proxy-Client-IP = {}", ip); |
|  | } |
|  | if(ip == null || ip.length() == 0 || "unknown".equalsIgnoreCase(ip)) { |
|  | ip = request.getHeader("WL-Proxy-Client-IP"); |
|  | logger.info("WL-Proxy-Client-IP = {}", ip); |
|  | } |
|  | if(ip == null || ip.length() == 0 || "unknown".equalsIgnoreCase(ip)) { |
|  | ip = request.getRemoteAddr(); |
|  | logger.info("RemoteAddr-IP = {}", ip); |
|  | } |
|  | if(!StringUtils.isEmpty(ip)) { |
|  | ip = ip.split(",")[0]; |
|  | } |
|  | return ip; |
|  | } |
|  | } |

public static String getIp(HttpServletRequest request){  
 String ip = request.getHeader("x-forwarded-for");  
 if (ip == null || ip.length() == 0 || "unknown".equalsIgnoreCase(ip)) {  
 ip = request.getHeader("X-Real-IP");  
 }  
 if (ip == null || ip.length() == 0 || "unknown".equalsIgnoreCase(ip)) {  
 ip = request.getRemoteAddr();  
 }  
 if(ip.toLowerCase().contains("x-forwarded-for")){  
 String ip\_temp = "";  
 if(ip.contains(":") && ip.split(":").length>0 && !ip.endsWith(":")){  
 ip\_temp = ip.split(":")[1];  
 if(StringUtils.isBlank(ip\_temp)) ip\_temp = ip.substring(0,ip.lastIndexOf("X-Forwarded-For"));  
 }else{  
 ip\_temp = ip.substring(0,ip.lastIndexOf("X-Forwarded-For"));  
 }  
 ip = ip\_temp.replaceAll(" ","");  
 }  
 if (ip.contains(",")) {  
 String ip\_temp = ip.split(",")[0];  
 ip\_temp = ip\_temp.replaceAll(" ", "");  
 if(ip\_temp.startsWith("10.") && ip.split(",").length>1){  
 ip = ip.split(",")[1];  
 ip = ip.replaceAll(" ", "");  
 }else ip = ip\_temp;  
 }  
 return ip;  
}