获得BeanInfo

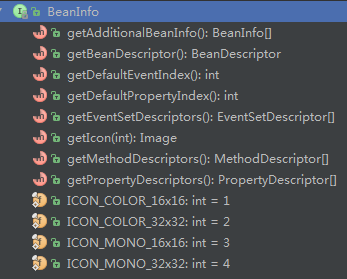
通过Introspector.*getBeanInfo*(beanClass)

两者均位于包 java.beans

前者是个接口

提供信息

explicit information about the methods,  
\* properties, events, etc, of their bean.



后者Introspector.*getBeanInfo*(Class<?>)

public static BeanInfo getBeanInfo(Class<?> beanClass)

通过Class获得一个BeanInfo对象

将对象序列化为字符串

org.apache.commons.lang3. SerializationUtils. serialize

(Object) 获得一个byte[]

public static byte[] serialize(final Serializable obj) {

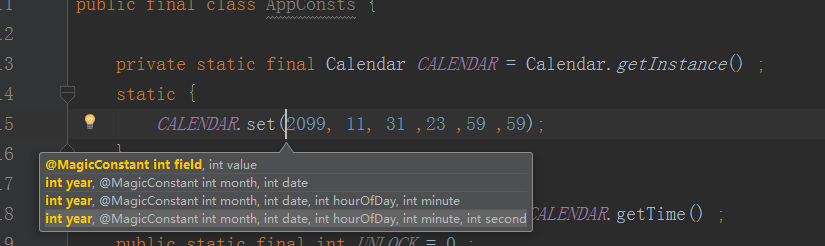
public static <T> T deserialize(final byte[] objectData)

实质是将对象通过ObjectInputStream读取对象写出到ByteArrayOutputStream实现序列化，

通过ByteArrayInputStream读取数组写出到ObjectinputStream再强转实现反序列化。

还是需要对象实现Serializable接口，不然报错。

Calendar快速设置时间，不必format



获得ip

public static String getIpAddr(HttpServletRequest request){  
   
 // 通过请求头获取ip地址  
 String ip = request.getHeader("x-forwarded-for");  
 // 判断ip地址是否是代理地址  
 if(ip == null || ip.length() == 0 || "unknown".equalsIgnoreCase(ip)) {  
 ip = request.getHeader("Proxy-Client-IP");  
 }  
 // 判断ip地址是否是代理地址  
 if (ip == null || ip.length() == 0 || "unknown".equalsIgnoreCase(ip)) {  
 ip = request.getHeader("WL-Proxy-Client-IP");  
 }  
 // 判断ip地址是否是代理地址  
 if (ip == null || ip.length() == 0 || "unknown".equalsIgnoreCase(ip)) {  
 ip = request.getRemoteAddr();  
 }  
 return ip;  
}

允许跨域

private void crossDomain(HttpServletRequest request,  
 HttpServletResponse response) throws MalformedURLException {  
 String referer = request.getHeader("Referer");  
 URL u = new URL(referer);   
 String host = u.getHost().toLowerCase();  
 // 如果域名包含.zhihuishu.com，则允许其跨域  
 if(StringUtils.*contains*(host, ".zhihuishu.com")) {  
// response.setHeader("Access-Control-Allow-Origin", referer);  
 response.setHeader("Access-Control-Allow-Origin", "\*");  
 response.setHeader("Access-Control-Allow-Methods", "POST, GET, OPTIONS, DELETE");  
 response.setHeader("Access-Control-Max-Age", "3600");  
 response.setHeader("Access-Control-Allow-Headers", "x-requested-with");  
 }  
 }