



链滴

java 复制非空对象属性值

作者: [zml2015](#)

原文链接: <https://ld246.com/article/1487063957683>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p>很多时候，我们需要通过对象拷贝，比如说VO类与数据库实体bean类、更新时非空对象不更新对同一对象不同数据分开存储等</p>

<p>用于对象拷贝，spring 和 Apache都提供了相应的工具类方法，BeanUtils.copyProperties</p>

<p>但是对于非空属性拷贝就需要自己处理了</p>

<p>在这里借用spring中org.springframework.beans.BeanUtils类提供的方法copyProperties(Object source, Object target, String... ignoreProperties) </p>

```
<pre class="brush: java">/**
 * Copy the property values of the given source bean into the given target bean,
 * ignoring the given "ignoreProperties".
 * &lt;p&gt;Note: The source and target classes do not have to match or even be derived
 * from each other, as long as the properties match. Any bean properties that the
 * source bean exposes but the target bean does not will silently be ignored.
 * &lt;p&gt;This is just a convenience method. For more complex transfer needs,
 * consider using a full BeanWrapper.
 * @param source the source bean
 * @param target the target bean
 * @param ignoreProperties array of property names to ignore
 * @throws BeansException if the copying failed
 * @see BeanWrapper
 */
 public static void copyProperties(Object source, Object target, String... ignoreProperties) th
ows BeansException {
     copyProperties(source, target, null, ignoreProperties);
```

```
/**
```

```
* Copy the property values of the given source bean into the given target bean.
```

```
* &lt;p&gt;Note: The source and target classes do not have to match or even be derived
```

```
* from each other, as long as the properties match. Any bean properties that the
```

```
* source bean exposes but the target bean does not will silently be ignored.
```

```
* @param source the source bean
```

```
* @param target the target bean
```

```
* @param editable the class (or interface) to restrict property setting to
```

```
* @param ignoreProperties array of property names to ignore
```

```
* @throws BeansException if the copying failed
```

```
* @see BeanWrapper
```

```
*/
```

```
private static void copyProperties(Object source, Object target, Class&lt;?
&gt; editable, String... ignoreProperties)
```

```
throws BeansException {
```

```
    Assert.notNull(source, "Source must not be null");
```

```
    Assert.notNull(target, "Target must not be null");
```



```

*/
public static String[] getNullPropertyNames(Object source) {
    final BeanWrapper src = new BeanWrapperImpl(source);
    java.beans.PropertyDescriptor[] pds = src.getPropertyDescriptors();

    Set<String> emptyNames = new HashSet<String>();
    for (java.beans.PropertyDescriptor pd : pds) {
        Object srcValue = src.getPropertyValue(pd.getName());
        if (srcValue == null)
            emptyNames.add(pd.getName());
    }
    String[] result = new String[emptyNames.size()];
    return emptyNames.toArray(result);
}

/**
 * @author 郑明亮
 * @Email zhengmingliang911@gmail.com
 * @Time 2017年2月14日 下午5:15:30
 * @Description <p>&gt; 拷贝非空对象属性值 </P>&gt;
 * @param source 源对象
 * @param target 目标对象
 */
public static void copyPropertiesIgnoreNull(Object source, Object target) {
    BeanUtils.copyProperties(source, target, getNullPropertyNames(source));
}

```


测试方法就不提供了，自行测试即可

 如果项目中使用的框架有Hibernate的话，则可以通过在实体类上添加下面两注解

```

@DynamicInsert(true)
@DynamicUpdate(true)

```

如果想对该注解进一步了解的话，那么可以去官网看英文文档，文档解释的很清楚，此不再赘述了

[https://www.mkyong.com/hibernate/hibernate-dynamic-insert-attribute-example/](https://www.mkyong.com/hibernate/hibernate-dynamic-insert-attribute-example/ "Hibernate – dynamic-insert attribute example")

