

```
1. package com.cn.copy.service;
2.
3. import java.lang.reflect.InvocationTargetException;
4. import java.util.Date;
5.
6. import org.apache.commons.beanutils.BeanUtils;
7. import org.apache.commons.beanutils.PropertyUtils;
8.
9. import com.cn.copy.vo.Middle;
10. import com.cn.copy.vo.MiddleDto;
11.
12. public class Test {
13.
14.     public static void testMapObjectToObject(int count) {
15.         Middle m1 = new Middle();
16.         m1.setFlag(true);
17.         m1.setId(1000L);
18.         m1.setItem(new int[] { 10, 20, 30 });
19.         m1.setPrice(12.33);
20.         m1.setType("hahaha!");
21.         m1.setDate(new Date());
22.         long t1 = System.currentTimeMillis();
23.         for (int i = 0; i < count; i++) {
24.             MiddleDto mg = new MiddleDto();
25.             VoPoConverter.copyProperties(mg, m1);
26.         }
27.         System.out.println("DozerBeanMapper-time1:"
28.             + (System.currentTimeMillis() - t1));
29.     }
30.
31.     public static void testPropertyCopyObjectObject(int
count) {
32.         Middle m1 = new Middle();
33.         m1.setFlag(true);
34.         m1.setId(1000L);
35.         m1.setItem(new int[] { 10, 20, 30 });
36.         m1.setPrice(12.33);
37.         m1.setType("hahaha!");
38.         m1.setDate(new Date());
39.         long t1 = System.currentTimeMillis();
40.
41.         for (int i = 0; i < count; i++) {
42.             MiddleDto mg = new MiddleDto();
43.             try {
```

```

44.         PropertyUtils.copyProperties(mg, m1);
45.     } catch (IllegalAccessException e) {
46.         // TODO Auto-generated catch block
47.         e.printStackTrace();
48.     } catch (InvocationTargetException e) {
49.         // TODO Auto-generated catch block
50.         e.printStackTrace();
51.     } catch (NoSuchMethodException e) {
52.         // TODO Auto-generated catch block
53.         e.printStackTrace();
54.     }
55. }
56. System.out.println("PropertyUtils-time2:"
57.     + (System.currentTimeMillis() - t1));
58. }
59.
60. public static void testBeanCopyObjectObject(int count) {
61.     Middle m1 = new Middle();
62.     m1.setFlag(true);
63.     m1.setId(1000L);
64.     m1.setItem(new int[] { 10, 20, 30 });
65.     m1.setPrice(12.33);
66.     m1.setType("hahaha!");
67.     m1.setDate(new Date());
68.     long t1 = System.currentTimeMillis();
69.
70.     for (int i = 0; i < count; i++) {
71.         MiddleDto mg = new MiddleDto();
72.         try {
73.             BeanUtils.copyProperties(mg, m1);
74.         } catch (IllegalAccessException e) {
75.             // TODO Auto-generated catch block
76.             e.printStackTrace();
77.         } catch (InvocationTargetException e) {
78.             // TODO Auto-generated catch block
79.             e.printStackTrace();
80.         }
81.     }
82.     System.out.println("BeanUtils-time3:"
83.         + (System.currentTimeMillis() - t1));
84. }
85.
86. public static void main(String[] args) {
87.     int total = 10000;
88.     Test.testMapObjectToObject(total);

```

```
89.         Test.testPropertyCopyObjectObject(total);
90.         Test.testBeanCopyObjectObject(total);
91.     }
92. }
```

1000次运行结果：

[\[html\] view plain copy](#)

```
1. DozerBeanMapper-time1:306
2. PropertyUtils-time2:30
3. BeanUtils-time3:42
```

100000次运行结果：

[\[html\] view plain copy](#)

```
1. DozerBeanMapper-time1:2904
2. PropertyUtils-time2:1155
3. BeanUtils-time3:2663
```

1000000次运行结果：

[\[html\] view plain copy](#)

```
1. DozerBeanMapper-time1:26588
2. PropertyUtils-time2:11454
3. BeanUtils-time3:26315
```

测试结果：

- 1、万次内的效率beanutils比dozer要快
- 2、十万次以上的效率beanutils的效率和dozer差不多。
- 3、以上是在只考虑基本类型的情况下比较的，dozer如果深拷贝或者关联拷贝时，效率应该会更慢。
- 4、PropertyUtils和BeanUtils的功能基本一致，唯一的区别是：**BeanUtils在对Bean赋值时会进行类型转化，而PropertyUtils不会对类型进行转化，如果类型不同则会抛出异常！**，这可以解释PropertyUtils效率比其他两种要高的原因。

建议：基本类型在源目标类型一致的情况下使用：**PropertyUtils效率会更高。**

复杂类型的拷贝可以使用：**Dozer。**