



黑客派

java 反射处理实体类转换

作者: [dengjunwen](#)

原文链接: <https://hacpai.com/article/1480563007164>

来源网站: [黑客派](#)

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

<p> 最近遇到这样的情况：系统B依赖于系统A，系统C依赖于系统B，但是系统C有些方法直接调用了系统A，所以这里系统C，跨级调用了。所以B系统C和A做了中介，将C要调用的方法转发了一下。</p>

```
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></script>
<!-- 黑客派PC帖子内嵌-展示 -->
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342" data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></ins>
<script>
    (adsbygoogle = window.adsbygoogle || []).push({});
</script>
<p><span><a href="https://link.hacpai.com/forward?goto=http%3A%2F%2F7xrj9d.com1.z0glb.cloudn.com%2FFp-ROHEVYsDJU7YxFSoPfGbhBnUr" class="fancybox" target="_blank" rel="nofollow ugc"></a></span></p>
<p>例子：<br>--</p>
<p>如：A系统中的 ResultDTO query(QueryParamDTO dto)，B系统需要重新写一个方法，然后用，并且将结果返回。以供C调用，这样就去掉了C对A的依赖。</p>
<p>那么问题来了。ResultDTO和QueryParamDTO这两个类型也是A系统中的。所以B系统还得重封装一个B自己的DTO (BResultDTO和BQueryParamDTO) 返回给C，于是就有了BQueryDTO转为QueryDTO的过程。</p>
<p><br>两个类生成的对象不能互相强制转换，所以只能一个个属性值设置。由于属性名称又相同所以就利用java反射来做。代码如下：其中遇到了枚举类型 (enum) 的转换的问题，不能直接set，要转换成定义的enum类型才可以。</p>
<p><br>如：CardTypeEnum类型直接设置为NocardTypeEnum，尽管里面的枚举元素都是相同都不行。具体转换看代码：</p>
<p>&nbsp;</p>
<pre class="prettyprint"> private void reflectSetValue(BasicParamDTO basicParamDTO){
    Method[] methods = basicParamDTO.getClass().getMethods();
    for (Method method : methods) {
        String fieldName = StringUtils.replaceOnce(method.getName(), "set", "");
        fieldName = StringUtils.uncapitalize(fieldName);
        Field sourceField = ReflectionUtils.findField(this.verifyParamOuterDTO.getClass(), fieldName);
        if (CheckUtils.isNull(sourceField)) {
            continue;
        }
        // 取得属性
        Field targetField = ReflectionUtils.findField(basicParamDTO.getClass(), fieldName);
        makeAccessible(targetField);
        makeAccessible(sourceField);
<pre><code class="highlight-chroma">        // 赋值给这个field
try {
    Object obj = sourceField.get(this.verifyParamOuterDTO);
    if (CheckUtils.isNull(obj)){
        continue;
    }
    //设置枚举类型，将一个枚举类，转换成另一个枚举类
    if (obj.getClass().isEnum()){
        Method valueOfMethod = targetField.getType().getMethod("valueOf", String.class);
        obj = ReflectionUtils.invokeJdbcMethod(valueOfMethod, null, new Object[]{obj.toString()});
    }
}</code></pre>
```

```
        }
        targetField.set(basicParamDTO, obj);
    } catch (Exception e) {
        LOGGER.info("反射赋值错误: {}", fieldName);
    }
}
/**
 * 强制转换fileld可访问.
 */
protected static void makeAccessible(final Field field) {
    if (!Modifier.isPublic(field.getModifiers()) || !Modifier.isPublic(field.getDeclaringClass().getModifiers())) {
        field.setAccessible(true);
    }
}
</code></pre>
</pre>
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></script>
<!-- 黑客派PC帖子内嵌-展示 --&gt;
&lt;ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342"
data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"&gt;&lt;/ins&gt;
&lt;script&gt;
    (adsbygoogle = window.adsbygoogle || []).push({});
&lt;/script&gt;
&lt;p&gt;&lt;span&gt;&ampnbsp;&lt;/span&gt;&lt;/p&gt;</pre>
```