JSONObject.toJavaObject 对首字母大写的属性设置失败

作者：wenyl

原文链接：https://ld246.com/article/1639106261613

来源网站：链滴

今天使用JSONObject.toJavaObject()方法试图将一个JSONObject对象装换为指定class对象，该对属性如下所示

```
private Integer ID;

    private String fertilizerSiteName;

    private String userName;

    private Boolean sendOk;
```

转换后发现ID字段一直为null，然后我们跟踪代码发现，fastjson通过获取set开头的方法，然后通过符串截取来获取属性(见com.alibaba.fastjson.util.DeserializeBeanInfo的DeserializeBeanInfo computeSetters(Class<?> clazz, Type type)方法)，因为正常的命名习惯都是set然后属性首字母大写，常这么写是没有问题的，不过这里我们的ID都是大写，他这里默认截取后首字母小写ID，就变成了iD来做映射的时候就找不到字段了，代码如下

```
if (methodName.startsWith("set")) {
                char c3 = methodName.charAt(3);
                String propertyName;
                if (Character.isUpperCase(c3)) {
                    if (TypeUtils.compatibleWithJavaBean) {
                        propertyName = TypeUtils.decapitalize(methodName.substring(3));
                    } else {
                        propertyName = Character.toLowerCase(methodName.charAt(3)) + methoName.substring(4);
                    }
                } else if (c3 == '_') {
                    propertyName = methodName.substring(4);
                } else if (c3 == 'f') {
                    propertyName = methodName.substring(3);
                } else {
```

原文链接：JSONObject.toJavaObject 对首字母大写的属性设置失败

```
                if (methodName.length() < 5 || !Character.isUpperCase(methodName.charAt(4))
{
                    continue;
                }

                propertyName = TypeUtils.decapitalize(methodName.substring(3));
            }

            Field field = TypeUtils.getField(clazz, propertyName);
            if (field == null && method.getParameterTypes()[0] == Boolean.TYPE) {
                String isFieldName = "is" + Character.toUpperCase(propertyName.charAt(0)) +
propertyName.substring(1);
                field = TypeUtils.getField(clazz, isFieldName);
            }

            if (field != null) {
                JSONField fieldAnnotation = (JSONField)field.getAnnotation(JSONField.class);
                if (fieldAnnotation != null) {
                    ordinal = fieldAnnotation.ordinal();
                    serialzeFeatures = SerializerFeature.of(fieldAnnotation.serialzeFeatures());
                    if (fieldAnnotation.name().length() != 0) {
                        propertyName = fieldAnnotation.name();
                        beanInfo.add(new FieldInfo(propertyName, method, field, clazz, type, ordi
al, serialzeFeatures));
                        continue;
                    }
                }
            }

            beanInfo.add(new FieldInfo(propertyName, method, (Field)null, clazz, type, ordina
, serialzeFeatures));
            TypeUtils.setAccessible(method);
        }
```

如下，ID被转成了iD

从代码中可以看到fastjson是根据TypeUtils.compatibleWithJavaBean属性（默认false）决定是否换首字母大小写，我们在代码中调用JSONObject.toJavaObject前将这个值设置为true即可

```
TypeUtils.compatibleWithJavaBean = true;
B_FertilizerSite b_fertilizerSite = JSONObject.toJavaObject(data, B_FertilizerSite.class);
```

此时他获取属性就会调用decapitalize方法如下，可见对于一个属性，如果首字母和第二个字母都是写，那么直接返回，否则还是会做首字母转小写处理

```
public static String decapitalize(String name) {
        if (name != null && name.length() != 0) {
            if (name.length() > 1 && Character.isUpperCase(name.charAt(1)) && Character.isUppe
Case(name.charAt(0))) {
                return name;
            } else {
                char[] chars = name.toCharArray();
                chars[0] = Character.toLowerCase(chars[0]);
                return new String(chars);
            }
        } else {
            return name;
        }
    }
```

原文链接：JSONObject.toJavaObject 对首字母大写的属性设置失败