

CGLib实现变化字段探测的供能 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/145/2021_2022_CGLib_E5_AE_9E_E7_8E_c104_145588.htm 为了巩固 CGLib 的知识，下面我们实现一个稍微复杂一点的例子。例、请实现一个拦截器，使其能够检测一个 JavaBean 的哪些字段改变了。（1）首先定义一个 JavaBean。

```
public class PersonInfo { private String name. private String email. private int age. private String address. public String getEmail() { return email. } public void setEmail(String email) { this.email = email. } public String getName() { return name. } public void setName(String name) { this.name = name. } public String getAddress() { return address. } public void setAddress(String address) { this.address = address. } public int getAge() { return age. } public void setAge(int age) { this.age = age. } }
```

（2）定义一个 MethodInterceptor，这一步是最关键的。

```
import java.lang.reflect.Method. import java.util.Collections. import java.util.HashSet. import java.util.Set. import net.sf.cglib.proxy.MethodInterceptor. import net.sf.cglib.proxy.MethodProxy. public class JavaBeanDataChangeInterceptor implements MethodInterceptor { private static final String SET = "set". private Set changedPropSet. public JavaBeanDataChangeInterceptor() { changedPropSet = new HashSet(). } public Object intercept(Object obj, Method method, Object[] args, MethodProxy proxy) throws Throwable { String name = method.getName(). if (name.startsWith(SET)) { String s = name.substring(SET.length()). changedPropSet.add(s). } return
```

```
proxy.invokeSuper(obj, args). } public Set getChangedPropSet() {  
return Collections.unmodifiableSet(changedPropSet). } public void  
reset() { changedPropSet.clear(). } }
```

 定义一个集合

changedPropSet 用来存放修改了的字段名，增加了一个方法
reset 用来清空此集合，增加了一个 getChangedPropSet 方法用
来供外界得到修改了的字段，为了防止调用者对
changedPropSet 做修改，因此我们采用

Collections.unmodifiableSet 对返回的集合进行不可修改的修饰。
在 intercept 方法中，我们判断如果被调用的方法以 set 开头，
则把此字段名放入 changedPropSet 集合中。100Test 下载频
道开通，各类考试题目直接下载。详细请访问

www.100test.com