

动态代理实现-----CGLib篇

作者: [crick77](#)

原文链接: <https://ld246.com/article/1462497182917>

来源网站: [链滴](#)

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

看了 @wanglei0622 的文章 [动态代理实现](#) 介绍了 JDK提供的方法，个人用CGLib比较多，所以补一下CGLib的动态代理实现，如果有错误，还请指出修正。

```
import net.sf.cglib.proxy.Enhancer;
import net.sf.cglib.proxy.MethodInterceptor;
import net.sf.cglib.proxy.MethodProxy;

import java.lang.reflect.Method;

/**
 * Created by CR on 2016/5/6.
 */
public class LogHandler implements MethodInterceptor {

    private static LogHandler instance = new LogHandler();

    private LogHandler(){

    }

    public static LogHandler getInstance(){
        return instance;
    }

    public <T> T getProxy(Class<T> cls) {
        return (T) Enhancer.create(cls, this);
    }

    @Override
    public Object intercept(Object o, Method method, Object[] objects, MethodProxy methodProxy) throws Throwable {
        doBefore();
        Object result = methodProxy.invokeSuper(o, objects);
        doAfter();
        return result;
    }

    private void doBefore(){
        System.out.println("do something before");
    }

    private void doAfter(){
        System.out.println("do something after");
    }

    public static void main(String[] args) {
        Eat proxy = LogHandler.getInstance().getProxy(Dog.class);
        proxy.eat();
    }
}
```

优势是，没有接口的类，也可以代理。Proxy.newProxyInstance这个方法参数也让人醉了