## 插件总述

插件原理：

在四大对象创建的时候

1. 每个创建出来的对象不是直接返回的，而是

interceptorChain.pluginAll(parameterHandler);

1. 获取到所有的Interceptor（拦截器）（插件需要实现的接口）；

调用interceptor.plugin(target);返回target包装后的对象

1. 插件机制，我们可以使用插件为目标对象创建一个代理对象；AOP（面向切面）

我们的插件可以为四大对象创建出代理对象；

代理对象就可以拦截到四大对象的每一个执行；

插件编写：

**① 编写Interceptor的实现类**

**② 使用@Intercepts注解完成插件签名**

**③ 将写好的插件注册到全局配置文件中**

**执行插件的顺序是配置的反序**

## 编写一个插件类

**完成插件签名：**

**告诉MyBatis当前插件用来拦截哪个对象的哪个方法（四大对象的任意方法）**

@Intercepts(

{

@Signature(type=StatementHandler.**class**,method="parameterize",args=java.sql.Statement.**class**)

})

**public** **class** MyFirstPlugin **implements** Interceptor{

**/\*\* intercept：拦截：**

**\* 拦截目标对象的目标方法的执行； \*/**

@Override

**public** Object intercept(Invocation invocation) **throws** Throwable {

System.***out***.println("MyFirstPlugin...intercept:"+invocation.getMethod());

**//动态的改变一下sql运行的参数：以前1号员工，实际从数据库查询3号员工**

Object target = invocation.getTarget();

System.***out***.println("当前拦截到的对象："+target);

//拿到：StatementHandler==>ParameterHandler===>parameterObject

//**拿到target的元数据**

MetaObject metaObject = SystemMetaObject.*forObject*(target);

Object value = metaObject.getValue("parameterHandler.parameterObject");

System.***out***.println("sql语句用的参数是："+value);

//**修改sql语句要用的参数**

metaObject.setValue("parameterHandler.parameterObject", 3);

//**执行目标方法**

Object proceed = invocation.proceed();

//**返回执行后的返回值**

**return** proceed;

}

/\*\* plugin：

\* 包装目标对象的：包装：为目标对象创建一个代理对象\*/

@Override

**public** Object plugin(Object target) {

//**我们可以借助Plugin的wrap方法来使用当前Interceptor包装我们目标对象**

System.***out***.println("MyFirstPlugin...plugin:mybatis将要包装的对象"+target);

Object wrap = Plugin.*wrap*(target, **this**);

//**返回为当前target创建的动态代理**

**return** wrap;

}

/\*\* setProperties：

\* 将插件注册时 的property属性设置进来（全局配置文件中）\*/

@Override

**public** **void** setProperties(Properties properties) {

System.***out***.println("插件配置的信息："+properties);

}

}

## 全局配置文件中配置

<!--plugins：注册插件 **interceptor，插件全类名** -->

<plugins>

**<plugin interceptor=*"com.atguigu.mybatis.dao.MyFirstPlugin"*>**

<property name=*"username"* value=*"root"*/>

<property name=*"password"* value=*"123456"*/>

</plugin>

</plugins>