



链滴

mybatis_plus 【Sql 注入器】

作者: [haxLook](#)

原文链接: <https://ld246.com/article/1617255917881>

来源网站: 链滴

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

MP的sql注入使用

我们已经知道，在MP中，通过AbstractSqlInjector将BaseMapper中的方法注入到了Mybatis容器这样这些方法才

可以正常执行。

那么，如果我们需要扩充BaseMapper中的方法，又该如何实现呢？

- 编写MyBaseMapper

```
import com.baomidou.mybatisplus.core.mapper.BaseMapper;
import java.util.List;
public interface MyBaseMapper<T> extends BaseMapper<T> {
    List<T> findAll();
}
```

其他的Mapper都可以继承该Mapper，这样实现了统一的扩展。

```
public interface UserMapper extends MyBaseMapperuser {
    User findById(Long id);
}
```

- 编写MySqlInjector

如果直接继承AbstractSqlInjector的话，原有的BaseMapper中的方法将失效，所以我们选择继承DefaultSqlInjector进行扩展。

```
import com.baomidou.mybatisplus.core.injector.AbstractMethod;
import com.baomidou.mybatisplus.core.injector.DefaultSqlInjector;
import java.util.List;
public class MySqlInjector extends DefaultSqlInjector {
    @Override
    public List<AbstractMethod> getMethodList() {
        List<AbstractMethod> methodList = super.getMethodList();
        methodList.add(new FindAll());
        // 再扩充自定义的方法
        list.add(new FindAll());
        return methodList;
    }
}
```

- 编写FindAll

```
import com.baomidou.mybatisplus.core.enums.SqlMethod;
import com.baomidou.mybatisplus.core.injector.AbstractMethod;
import com.baomidou.mybatisplus.core.metadata.TableInfo;
import org.apache.ibatis.mapping.MappedStatement;
import org.apache.ibatis.mapping.SqlSource;
public class FindAll extends AbstractMethod {
    @Override
    public MappedStatement injectMappedStatement(Class<?> mapperClass, Class<?>
```

```
modelClass, TableInfo tableInfo) {
String sqlMethod = "findAll";
String sql = "select * from " + tableInfo.getTableName();
SqlSource sqlSource = languageDriver.createSqlSource(configuration, sql,
modelClass);
return this.addSelectMappedStatement(mapperClass, sqlMethod, sqlSource,
modelClass, tableInfo);
}
}
```

- 注册到Spring容器

```
/**
 * 自定义SQL注入器
 */
@Bean
public MySqlInjector mySqlInjector(){
return new MySqlInjector();
}
```

总结: sql注入是用的不是很多, 主要是在MP中baseMapper自带的sql操作不满足你的业务要求, 而这一个sql操作在大部分的业务表中需要使用, 这个时候就需要自己编写一个自定义的sql注入到MP中