Atitit mybatis快速开发 的sql api接口

[1.1. sql模式 开发速度大大快与 映射模式 1](#_Toc4950)

[1.2. MyBatis Mapper 1](#_Toc17931)

[1.2.1. 代码 2](#_Toc16452)

[1.2.2. 原理 2](#_Toc14012)

[1.3. 参考资料 3](#_Toc3928)

## sql模式 开发速度大大快与 映射模式

## MyBatis Mapper

可能有些人也有过类似需求，一般都会选择使用其他的方式如Spring-JDBC等方式解决。

能否通过MyBatis实现这样的功能呢？

为了让通用Mapper更彻底的支持多表操作以及更灵活的操作，在<b>2.2.0版本</b>增加了一个可以直接执行SQL的新类SqlMapper。

注：3.3.0版本去掉了这个类，这个类现在在[EntityMapper项目](http://git.oschina.net/free/EntityMapper)

EntityMapper是通用Mapper2.x版本中的一部分，通用Mapper3.x以后将EntityMapper移出了通用Mapper，所以EntityMapper独立出来。

建议使用通用Mapper，通用Mapper3更强大，通用方法更多，更方便扩展。

### 代码

**import** com.github.abel533.sql.SqlMapper;

|  |
| --- |
| org.mybatis.spring.**SqlSessionFactoryBean** **factoryBean**=**SpringUtil**.*build4wash*().*getBeanV2*(**SqlSessionFactoryBean**.**class**);  SqlSessionFactory **sql\_sessionFactory** = factoryBean.getObject();  SqlSession **sqlSession** = sql\_sessionFactory.openSession();  //创建sqlMapper  **SqlMapper** **sqlMapper** = **new** SqlMapper(sqlSession);  //查询，返回List<Map>  List<Map<String, Object>> **list** = sqlMapper.selectList("select \* from s\_member limit 3");  **System**.***out***.println(**AtiJson**.*toJson*(list)); |

### 原理

public List<Map<String, Object>> selectList(String sql) {

String msId = msUtils.select(sql);

return sqlSession.selectList(msId);

}

Msid SELECT.-1361880592

/palmWin/src/main/java/com/github/abel533/sql/SqlMapper.java

private String select(String sql) {

String msId = newMsId(sql, SqlCommandType.SELECT);

if (hasMappedStatement(msId)) {

return msId;

}

StaticSqlSource sqlSource = new StaticSqlSource(configuration, sql);

newSelectMappedStatement(msId, sqlSource, Map.class);

return msId;

}

StaticSqlSource是mybatis提供的类库

## 参考资料

MyBatis直接执行SQL的工具SqlMapper - abel533的个人页面.html

MyBatis直接执行SQL的工具SqlMapper - abel533的个人页面.html