

一、Mybatis-Plus介绍

MyBatis-Plus(简称 MP),是一个 MyBatis 的增强工具包,只做增强不做改变. 为简化开发工作、提高生产率而生 我们的愿景是成为 Mybatis 最好的搭档,就像 魂斗罗 中的 1P、2P, 基友搭配, 效率翻倍。

Mybatis-Plus 的集成非常简单,对于 Spring,我们仅仅需要把 Mybatis 自带的 MybatisSqlSessionFactoryBean 替换为 MP 自带的即可。

```
<bean id="sqlSessionFactoryBean"
class="com.baomidou.mybatisplus.spring.MybatisSqlSessionFactoryBean">
    <!-- 数据源 -->
    <property name="dataSource" ref="dataSource"></property>
    <property name="configLocation" value="classpath:mybatis-config.xml"></property>
    <!-- 别名处理 -->
    <property name="typeAliasesPackage" value="zh1cheung.mp.beans"></property>

    <!-- 注入全局MP策略配置 -->
    <property name="globalConfig" ref="globalConfiguration"></property>
</bean>
```

二、入门

1) 提出问题: 假设我们已存在一张 tbl_employee 表, 且已有对应的实体类 Employee, 实现tbl_employee 表的 CRUD 操作我们需要做什么呢? 2) 实现方式: 基于 Mybatis 需要编写 EmployeeMapper 接口, 并手动编写 CRUD 方法提供 EmployeeMapper.xml 映射文件, 并手动编写每个方法对应的 SQL 语句. 基于 MP 只需要创建 EmployeeMapper 接口, 并继承 BaseMapper 接口. 这就是使用 MP 需要完成的所有操作, 甚至不需要创建 SQL 映射文件。

- 原理分析

1) 问题: xxxMapper 继承了 BaseMapper, BaseMapper 中提供了通用的 CRUD 方法,方法来源于 BaseMapper, 有方法就必须有 SQL, 因为 MyBatis 最终还是需要通过SQL 语句操作数据.

A. employeeMapper 的本质 org.apache.ibatis.binding.MapperProxy B. MapperProxy 中 sqlSession -> SqlSessionFactory C. SqlSessionFactory 中 → Configuration → MappedStatements 每一个 mappedStatement 都表示 Mapper 接口中的一个方法与 Mapper 映射文件中的一个 SQL。MP 在启动就会挨个分析 xxxMapper 中的方法, 并且将对应的 SQL 语句处理好, 保存到 configuration 对象中的 mappedStatements 中。

Configuration: MyBatis 或者 MP 全局配置对象 MappedStatement: 一个 MappedStatement 对象对应 Mapper 配置文件中的一个select/update/insert/delete 节点，主要描述的是一条 SQL 语句 SqlMethod : 枚举对象，MP 支持的 SQL 方法 TableInfo: 数据库表反射信息， 可以获取到数据库表相关的信息 SqlSource: SQL 语句处理对象 MapperBuilderAssistant: 用于缓存、SQL 参数、查询方剂结果集处理等.通过 MapperBuilderAssistant 将每一个 mappedStatement 添加到 configuration 中的 mappedstatements 中

三、条件构造器EntityWrapper

1) Mybatis-Plus 通过 EntityWrapper（简称 EW，MP 封装的一个查询条件构造器）或者Condition（与 EW 类似）来让用户自由的构建查询条件，简单便捷，没有额外的负担，能够有效提高开发效率 2) 实体包装器，主要用于处理 sql 拼接，排序，实体参数查询等 3) 注意: 使用的是**数据库字段**，不是 Java 属性! 4) 条件参数说明:

查询方式	说明
setSqlSelect	设置 SELECT 查询字段
where	WHERE 语句，拼接 + WHERE 条件
and	AND 语句，拼接 + AND 字段=值
andNew	AND 语句，拼接 + AND (字段=值)
or	OR 语句，拼接 + OR 字段=值
orNew	OR 语句，拼接 + OR (字段=值)
eq	等于=
allEq	基于 map 内容等于=
ne	不等于<>
gt	大于>
ge	大于等于>=
lt	小于<
le	小于等于<=
like	模糊查询 LIKE
notLike	模糊查询 NOT LIKE
in	IN 查询
notin	NOT IN 查询
isNull	NULL 值查询
isNotNull	IS NOT NULL
groupBy	分组 GROUP BY
having	HAVING 关键词
orderBy	排序 ORDER BY
orderAsc	ASC 排序 ORDER BY
orderDesc	DESC 排序 ORDER BY
exists	EXISTS 条件语句
notExists	NOT EXISTS 条件语句
between	BETWEEN 条件语句
notBetween	NOT BETWEEN 条件语句
addFilter	自由拼接 SQL
last	拼接在最后，例如：last("LIMIT 1")

```
List<Employee> userList = employeeMapper.selectPage(new Page<Employee>(2, 3), new
EntityWrapper<Employee>()
.eq("last_name", "MyBatisPlus")
.eq("gender", 1)
.between("age", 18, 50))
```

```
List<Employee> userListCondition = employeeMapper.selectPage(
new Page<Employee>(2, 3), Condition.create()
.eq("gender", 1)
.eq("last_name", "MyBatisPlus")
.between("age", 18, 50));
```

四、ActiveRecord(活动记录)

Active Record(活动记录), 是一种**领域模型模式**, 特点是一个模型类对应关系型数据库中的一个表, 而模型类的一个实例对应表中的一行记录。

仅仅需要让实体类继承 Model 类且实现主键指定方法, 即可开启 AR 之旅。

```
@TableName("tbl_employee")
public class Employee extends Model<Employee>{
// .. fields
// .. getter and setter
@Override
protected Serializable pkVal() {
return this.id;
}
```

五、代码生成器

- MP 的代码生成器都是基于 java 代码来生成。MBG 基于 xml 文件进行代码生成
- MyBatis 的代码生成器可生成: 实体类、Mapper 接口、Mapper 映射文件
- MP 的代码生成器可生成: 实体类(可以选择是否支持 AR)、Mapper 接口、Mapper 映射文件、Service 层、Controller 层。
- MP 的代码生成器默认使用的是 Apache 的 Velocity 模板

六、插件扩展

四大对象的每个对象在创建时，都会执行 `interceptorChain.pluginAll()`，会经过每个插件对象的 `plugin()` 方法，目的是为当前的四大对象创建代理。代理对象就可以拦截到四大对象相关方法的执行，因为要执行四大对象的方法需要经过代理

七、自定义全局操作

根据 MybatisPlus 的 **AutoSqlInjector** 可以自定义各种你想要的 sql，注入到全局中，相当于自定义 Mybatisplus 自动注入的方法。

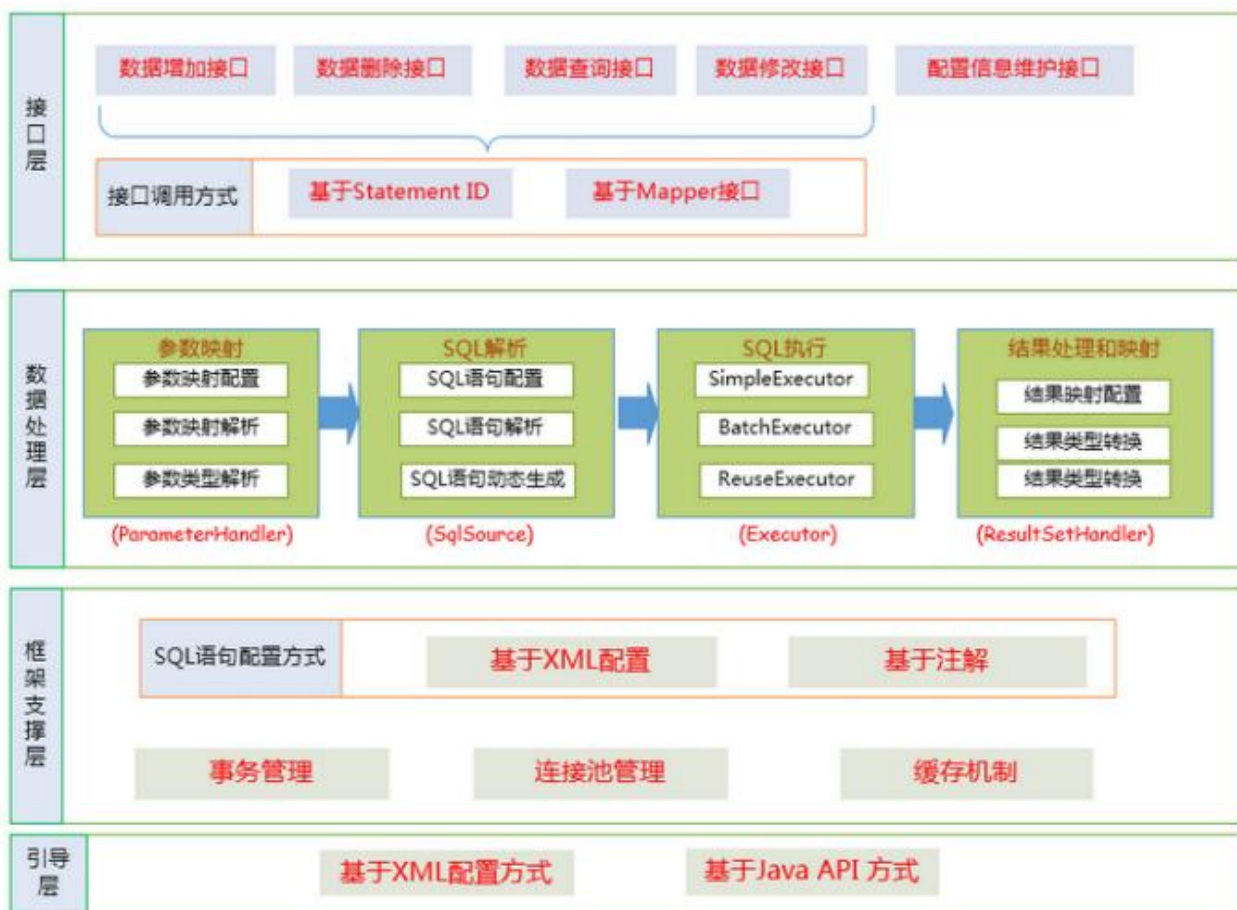
之前需要在 xml 中进行配置的 SQL 语句，现在通过扩展 `AutoSqlInjector` 在加载 mybatis 环境时就注入。

- 1) 在 Mapper 接口中定义相关的 CRUD 方法
- 2) 扩展 `AutoSqlInjector inject` 方法，实现 Mapper 接口中方法要注入的 SQL
- 3) 在 MP 全局策略中，配置自定义注入器

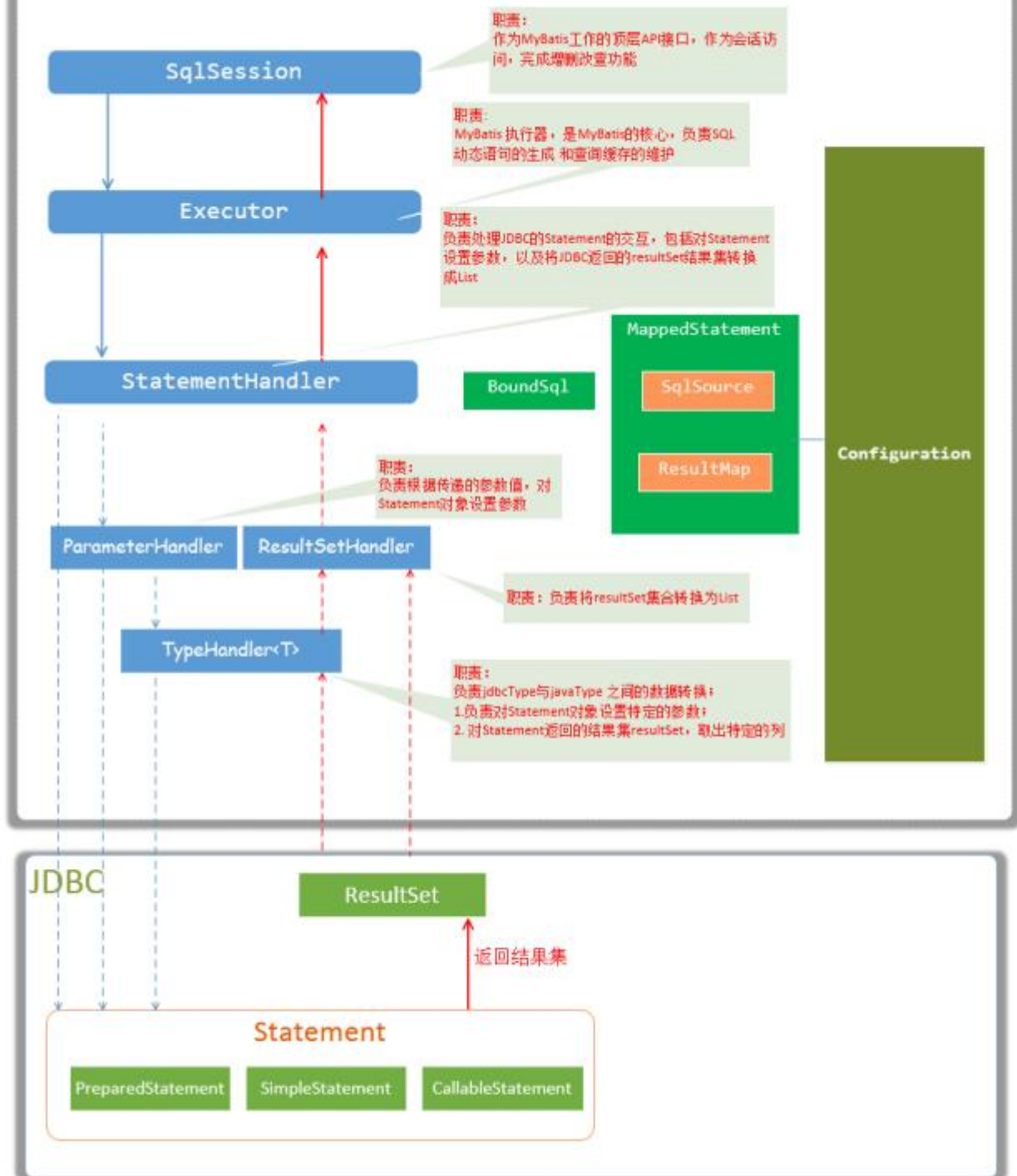
八、公共字段自动填充

- 1) 注解填充字段 `@TableField(fill = FieldFill.INSERT)` 查看 `FieldFill`
- 2) 自定义公共字段填充处理器
- 3) MP 全局注入 自定义公共字段填充处理器

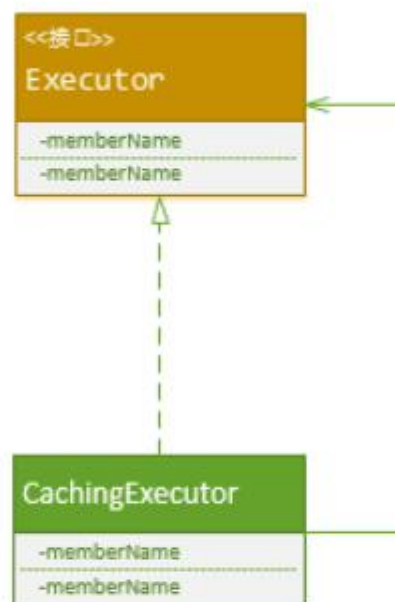
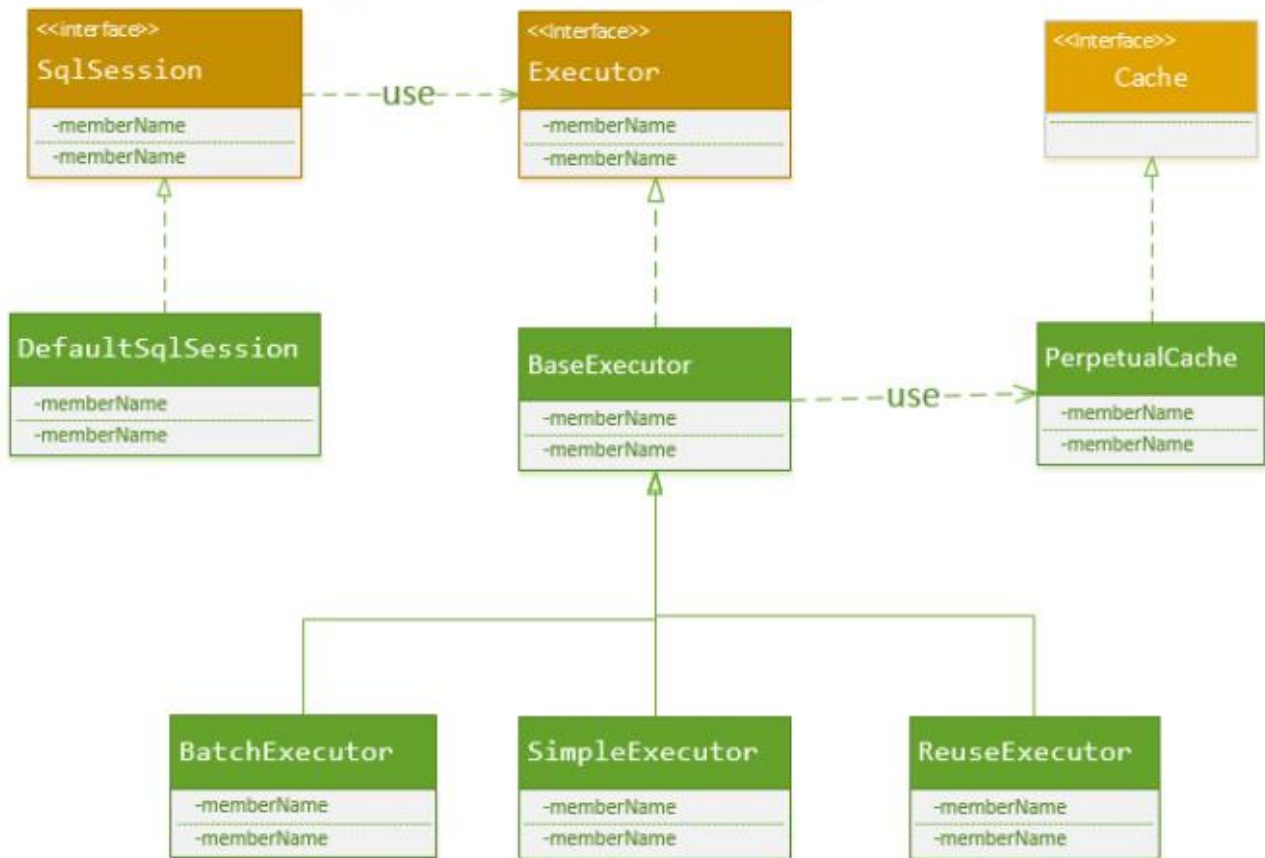
mybatis架构图



MyBatis层次结构



SqlSession和Executor以及Cache的关系类图



运用decorator模式，使用CachingExecutor对Executor的功能进行了增强，增加了二级缓存的功能