MyBatis-Plus

一、简介

1.1 是什么

MyBatis-Plus：为简化开发而生，是MyBatis增强版

只做增强不做改变，引入它不会对现有工程产生影响，只需简单配置，即可快速进行 CRUD 操作，从而节省大量时间，热加载、代码生成、分页、性能分析等功能一应俱全。

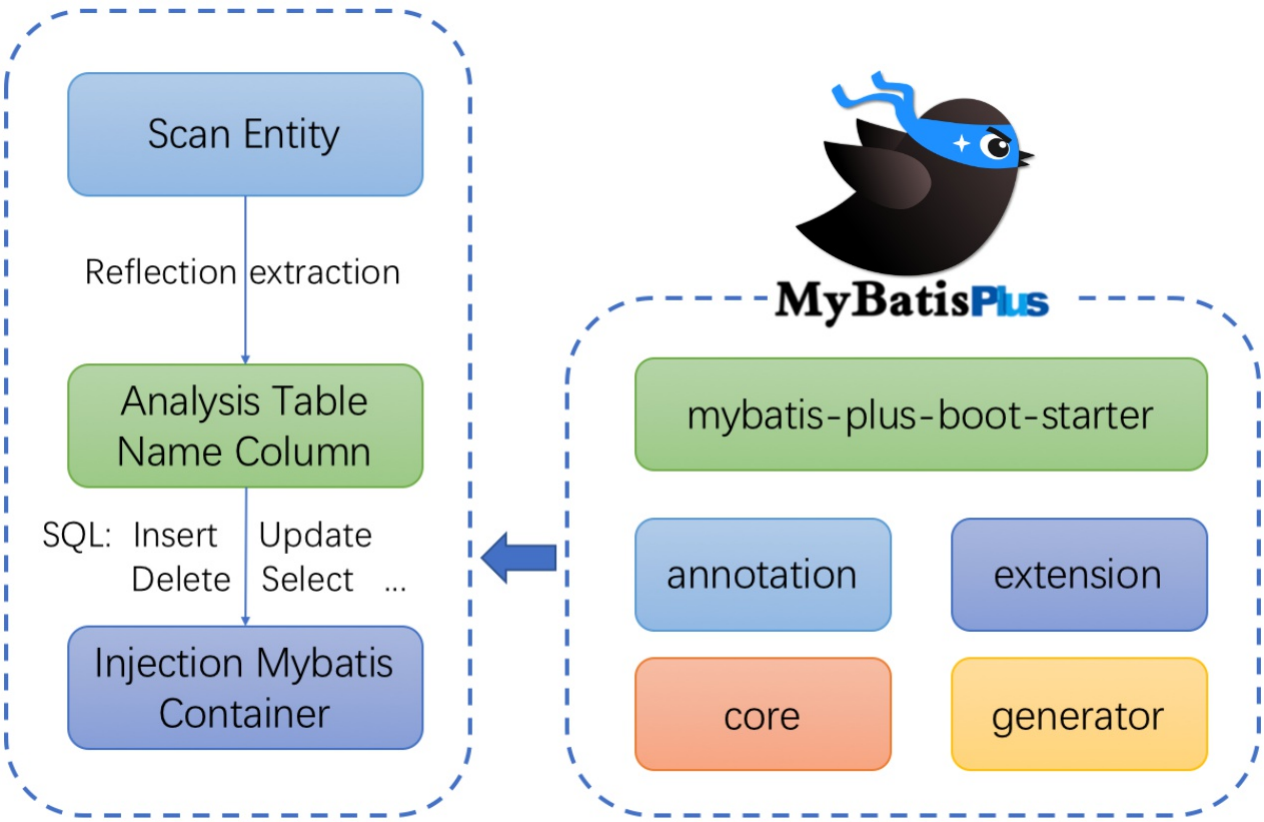
MyBatis-Plus（简称 MP）是一个 MyBatis的增强工具，在 MyBatis 的基础上只做增强不做改变，为简化开发、提高效率而生。

官网地址：https://mp.baomidou.com/

1.2 MP特性

- **无侵入**：只做增强不做改变，引入它不会对现有工程产生影响，如丝般顺滑
- **损耗小**：启动即会自动注入基本 CURD，性能基本无损耗，直接面向对象操作
- **强大的 CRUD 操作**：内置通用 Mapper、通用 Service，仅仅通过少量配置即可实现单表大部分 CRUD 操作，更有强大的条件构造器，满足各类使用需求
- **支持 Lambda 形式调用**：通过 Lambda 表达式，方便的编写各类查询条件，无需再担心字段写错
- **支持主键自动生成**：支持多达 4 种主键策略（内含分布式唯一 ID 生成器 - Sequence），可自由配置，完美解决主键问题
- **支持 ActiveRecord 模式**：支持 ActiveRecord 形式调用，实体类只需继承 Model 类即可进行强大的 CRUD 操作
- **支持自定义全局通用操作**：支持全局通用方法注入（ Write once, use anywhere ）
- **内置代码生成器**：采用代码或者 Maven 插件可快速生成 Mapper 、 Model 、 Service 、 Controller 层代码，支持模板引擎，更有超多自定义配置等您来使用
- **内置分页插件**：基于 MyBatis 物理分页，开发者无需关心具体操作，配置好插件之后，写分页等同于普通 List 查询
- **分页插件支持多种数据库**：支持 MySQL、MariaDB、Oracle、DB2、H2、HSQL、SQLite、Postgre、SQLServer2005、SQLServer 等多种数据库
- **内置性能分析插件**：可输出 Sql 语句以及其执行时间，建议开发测试时启用该功能，能快速揪出慢查询
- **内置全局拦截插件**：提供全表 delete 、 update 操作智能分析阻断，也可自定义拦截规则，预防误操作

1.3 MP结构



二、初体验

2.1 准备数据库

现有一张 User 表，其表结构如下：

id name age email 1 Jone 18 test1@baomidou.com 2 Jack 20 test2@baomidou.com 3 Tom 28 test3@baomidou.com 4 Sandy 21 test4@baomidou.com 5 Billie 24 test5@baomidou.com

``

```
CREATE TABLE user
(
    id BIGINT(20) NOT NULL COMMENT '主键ID',
    name VARCHAR(30) NULL DEFAULT NULL COMMENT '姓名',
    age INT(11) NULL DEFAULT NULL COMMENT '年龄',
    email VARCHAR(50) NULL DEFAULT NULL COMMENT '邮箱',
    PRIMARY KEY (id)
);
```

```
INSERT INTO user (id, name, age, email) VALUES (1, 'Jone', 18, 'test1@baomidou.com'), (2, 'Jack', 20, 'test2@baomidou.com'), (3, 'Tom', 28, 'test3@baomidou.com'), (4, 'Sandy', 21, 'test4@baomidou.com'), (5, 'Billie', 24, 'test5@baomidou.com');
```

2.2 创建SpringBoot项目

2.3 依赖jar

``

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <scope>runtime</scope>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <optional>true</optional>
</dependency>
<dependency>
    <groupId>com.baomidou</groupId>
    <artifactId>mybatis-plus-boot-starter</artifactId>
    <version>3.1.2</version>
</dependency>
<dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>druid</artifactId>
    <version>1.1.10</version>
</dependency>
<!-- https://mvnrepository.com/artifact/io.springfox/springfox-swagger2 -->
<dependency>
    <groupId>io.springfox</groupId>
    <artifactId>springfox-swagger2</artifactId>
    <version>2.9.2</version>
</dependency>
<!-- https://mvnrepository.com/artifact/io.springfox/springfox-swagger2 -->
<dependency>
    <groupId>io.springfox</groupId>
    <artifactId>springfox-swagger-ui</artifactId>
    <version>2.9.2</version>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-logging</artifactId>
</dependency>
```

2.4 代码编写

实体层

``

```
@Data
@TableName("user")
public class User {
    @TableId(type = IdType.AUTO)
    private Long id;
    private String name;
    private Integer age;
    private String email;
    public User() {
    }

    public User(String name, Integer age, String email) {
        this.name = name;
        this.age = age;
        this.email = email;
    }
}
```

持久层

``

```
public interface UserDao extends BaseMapper<User> {

    int insertCount(List<User> dtlist);
}
```

业务逻辑层

``

```
public interface UserService extends IService<User> {
    int saveCount(int count);
}
```

``

```
@Service
public class UserServiceImpl extends ServiceImpl<UserDao,User> implements UserService {
    @Override
    public int saveCount(int count) {
        List<User> userList=new ArrayList<>();
        for(int i=1;i<=count;i++){
            userList.add(new User("CEO-"+i,i*2,"ceo"+i+"@163.com"));
        }
        return  getBaseMapper().insertCount(userList);
    }
}
```

控制层

``

```
@RestController
public class UserController {
    @Autowired
    private UserService userService;
    private Logger logger=LoggerFactory.getLogger(UserController.class);
    @PostMapping("/api/user/add.do")
    public boolean save(@RequestBody User user, HttpServletRequest request){
        logger.info(request.getRemoteAddr()+":发来请求-用户数据新增");
        return userService.save(user);

    }
    @PutMapping("/api/user/update.do")
    public boolean update(@RequestBody User user){
        //QueryWrapper 条件拼接
        logger.error("条件拼接修改错误");
        return userService.update(user,new QueryWrapper<User>().eq("id",user.getId()));
    }
    @DeleteMapping("/api/user/del/{id}")
    public boolean del(@PathVariable int id){
        logger.warn("执行了删除: "+id);
        return userService.removeById(id);
    }
    @GetMapping("/api/user/all.do")
    public List<User> all(HttpServletRequest request){

        logger.info(request.getRemoteAddr()+":查询了全部数据");
        return userService.list();
    }
    @GetMapping("/api/user/page/{page}/{count}")
    public List<User> page(@PathVariable int page,@PathVariable int count){
        Page<User> page1=new Page<>(page,count);
        return userService.page(page1).getRecords();
    }
    @PostMapping("/api/user/savecount/{count}")
    public int saveAll(@PathVariable int count){
        return userService.saveCount(count);
    }
}
```

详细源码地址:

三、核心

3.1 自定义方法

dao中直接定义方法

service中getBaseMapper()

3.2 条件

AbstractWrapper

```
QueryWrapper
```

3.3 分页

```
Page<User> page1=new Page<>(page,count);


    /**
     * 分页插件
     */
    @Bean
    public PaginationInterceptor paginationInterceptor() {
        PaginationInterceptor paginationInterceptor = new PaginationInterceptor();
        // paginationInterceptor.setLimit(你的最大单页限制数量，默认 500 条，小于 0 如 -1 不受限
制);
        return paginationInterceptor;
    }
```

3.4 xml文件的支持

```
mybatis-plus:
  mapper-locations: classpath*:/mapper/*.xml
```