

一、简介

1、官网

<http://mp.baomidou.com>

2、特点

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。

愿景

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。



润物无声

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。

效率至上

只需简单配置，即可快速进行单表 CRUD 操作，从而节省大量时间。

丰富功能

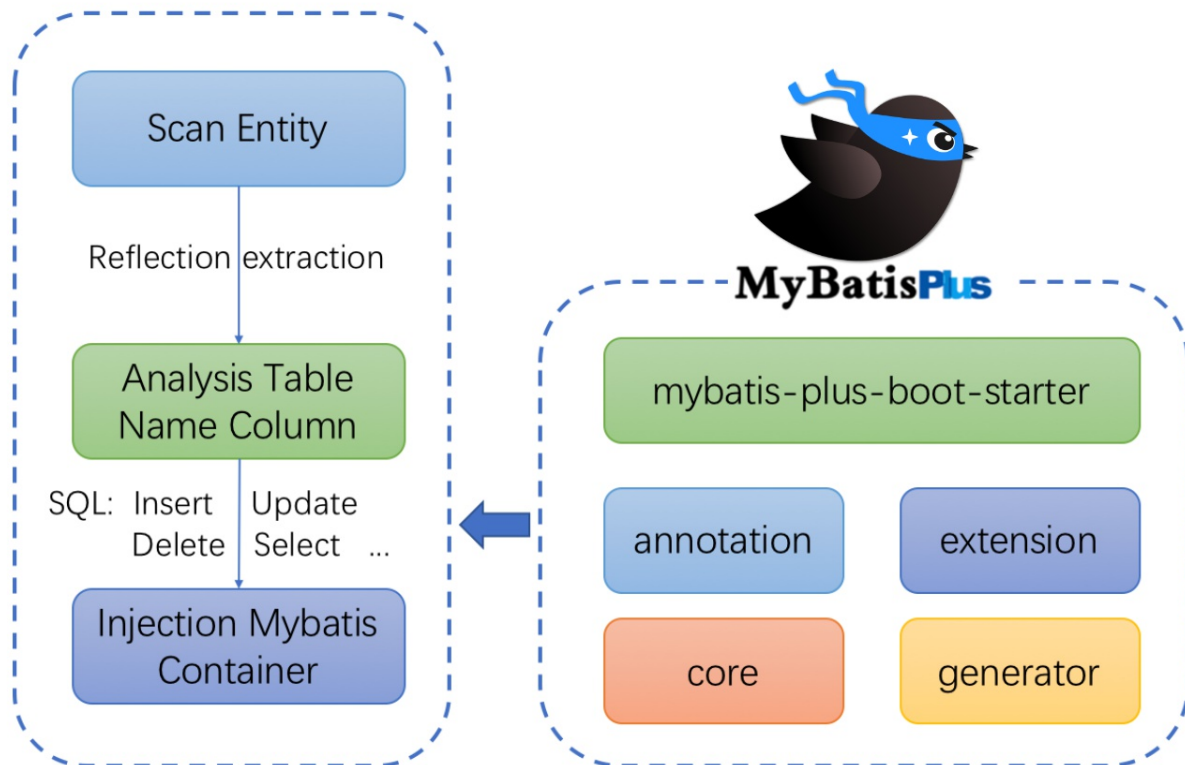
代码生成、物理分页、性能分析等功能一应俱全。

3、支持数据库

mysql、mariadb、oracle、db2、h2、hsqldb、sqlite、postgresql、sqlserver、presto、Gauss、Firebird

Phoenix、clickhouse、Sybase ASE、OceanBase、达梦数据库、虚谷数据库、人大金仓数据库、南大通用数据库、

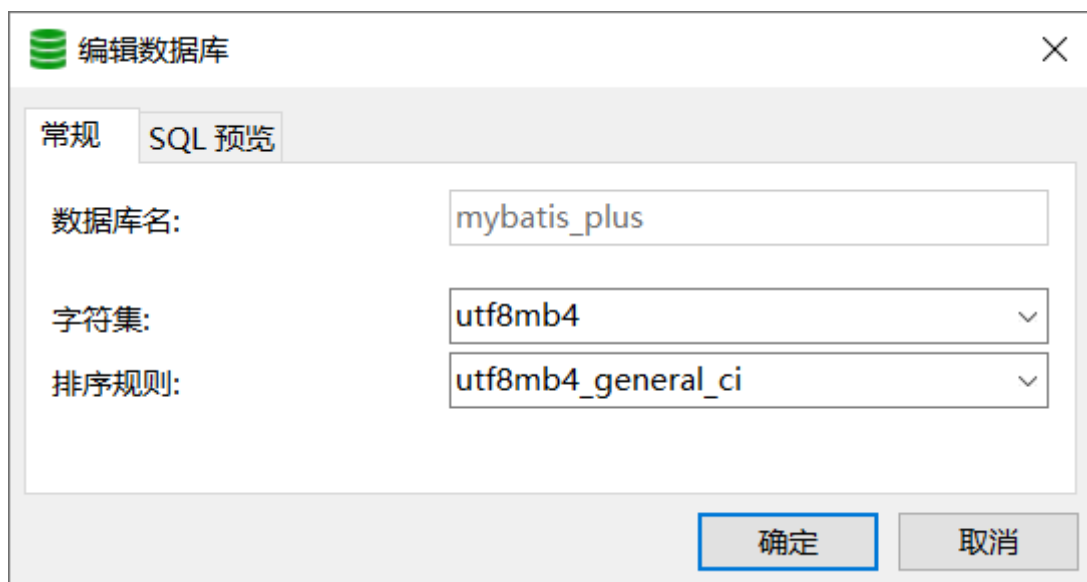
4、框架结构



二、快速入门

1、创建数据库

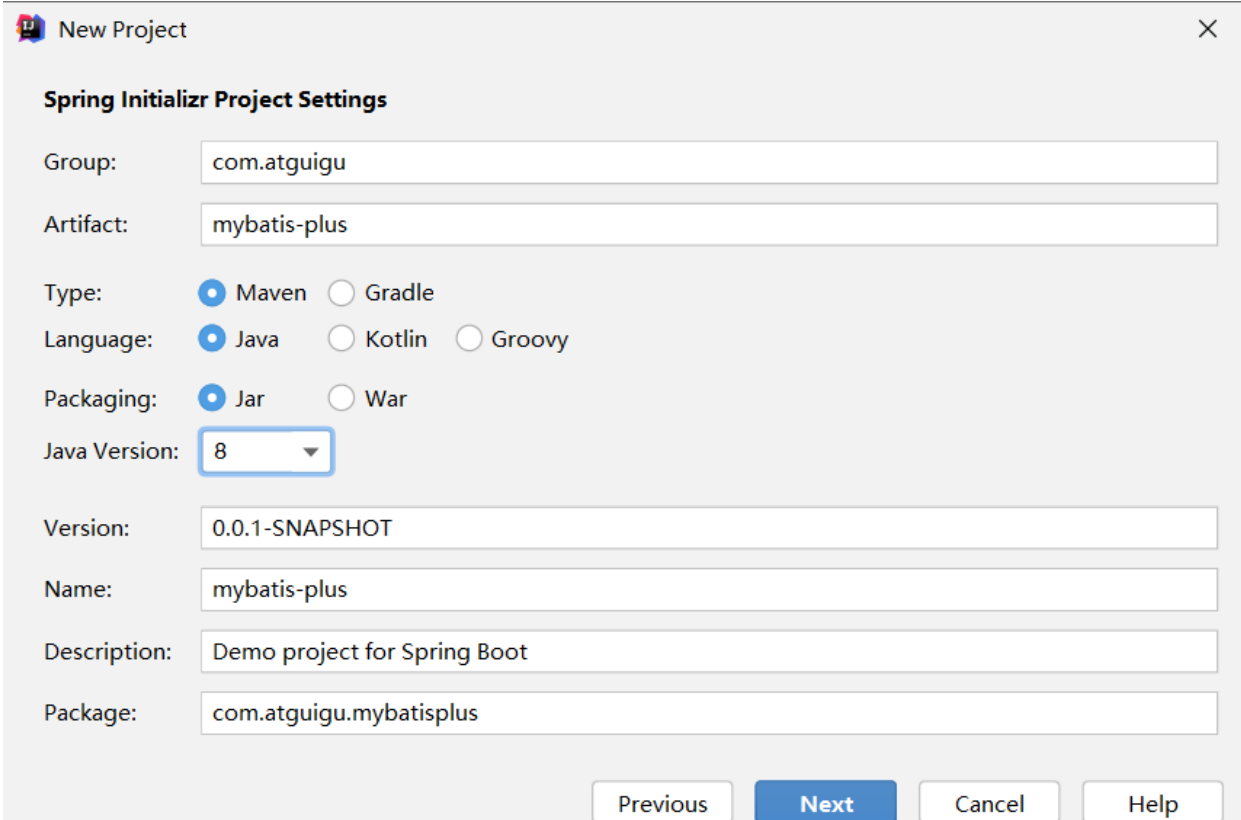
创建数据库：mybatis_plus



创建数据表：user

```
1 CREATE TABLE user
2 (
3     id BIGINT(20) NOT NULL COMMENT '主键ID',
4     name VARCHAR(30) NULL DEFAULT NULL COMMENT '姓名',
5     age INT(11) NULL DEFAULT NULL COMMENT '年龄',
6     email VARCHAR(50) NULL DEFAULT NULL COMMENT '邮箱',
7     PRIMARY KEY (id)
8 );
9 INSERT INTO user (id, name, age, email) VALUES
10 (1, 'Jone', 18, 'test1@baomidou.com'),
11 (2, 'Jack', 20, 'test2@baomidou.com'),
12 (3, 'Tom', 28, 'test3@baomidou.com'),
13 (4, 'Sandy', 21, 'test4@baomidou.com'),
14 (5, 'Billie', 24, 'test5@baomidou.com');
```

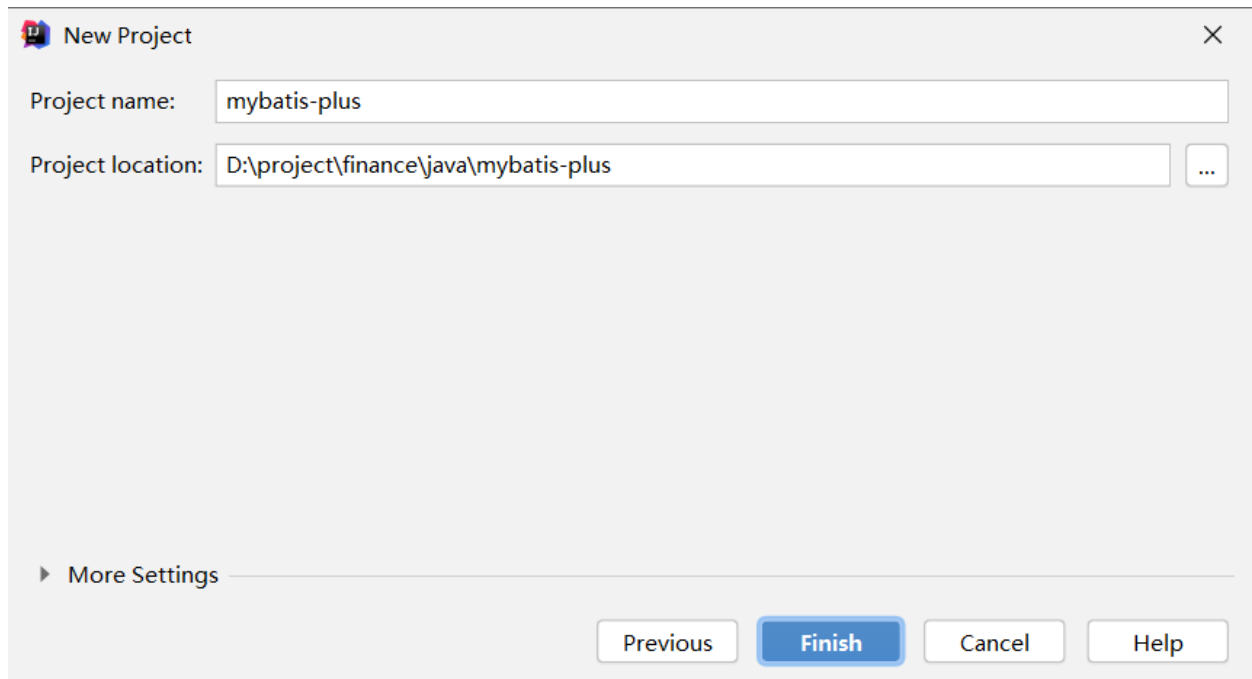
2、创建SpringBoot工程



The image shows the 'New Project' dialog box in an IDE, specifically the 'Spring Initializr Project Settings' tab. The dialog is titled 'New Project' and has a close button (X) in the top right corner. It contains several fields and options for configuring a new Spring Boot project:

- Group:** com.atguigu
- Artifact:** mybatis-plus
- Type:** ☒ Maven ☐ Gradle
- Language:** ☒ Java ☐ Kotlin ☐ Groovy
- Packaging:** ☒ Jar ☐ War
- Java Version:** 8 (selected from a dropdown menu)
- Version:** 0.0.1-SNAPSHOT
- Name:** mybatis-plus
- Description:** Demo project for Spring Boot
- Package:** com.atguigu.mybatisplus

At the bottom right, there are four buttons: 'Previous', 'Next' (highlighted in blue), 'Cancel', and 'Help'.



3、引入依赖

注意：SpringBoot版本修改为2.3.4

```
1 <dependencies>
2     <dependency>
3         <groupId>org.springframework.boot</groupId>
4         <artifactId>spring-boot-starter</artifactId>
5     </dependency>
6
7     <dependency>
8         <groupId>com.baomidou</groupId>
9         <artifactId>mybatis-plus-boot-starter</artifactId>
10        <version>3.4.1</version>
11    </dependency>
12
13    <dependency>
14        <groupId>mysql</groupId>
15        <artifactId>mysql-connector-java</artifactId>
16        <scope>runtime</scope>
17    </dependency>
18
19    <dependency>
20        <groupId>org.projectlombok</groupId>
21        <artifactId>lombok</artifactId>
```

```

22     <optional>true</optional>
23 </dependency>
24
25 <dependency>
26     <groupId>org.springframework.boot</groupId>
27     <artifactId>spring-boot-starter-test</artifactId>
28     <scope>test</scope>
29     <exclusions>
30         <exclusion>
31             <groupId>org.junit.vintage</groupId>
32             <artifactId>junit-vintage-engine</artifactId>
33         </exclusion>
34     </exclusions>
35 </dependency>
36 </dependencies>

```

4、修改配置文件

在 application.properties 配置文件中添加 MySQL 数据库的相关配置：

```

1 #mysql数据库连接
2 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
3 spring.datasource.url=jdbc:mysql://localhost:3306/mybatis_plus?serverTimezone=
4 spring.datasource.username=root
5 spring.datasource.password=123456

```

注意：如果定义了mysql驱动的依赖的版本为5，例如

```

1 <version>5.1.47</version>

```

则数据库连接配置为

```

1 #mysql数据库连接
2 spring.datasource.driver-class-name=com.mysql.jdbc.Driver
3 spring.datasource.url=jdbc:mysql://localhost:3306/mybatis_plus?characterEncoc
4 spring.datasource.username=root

```

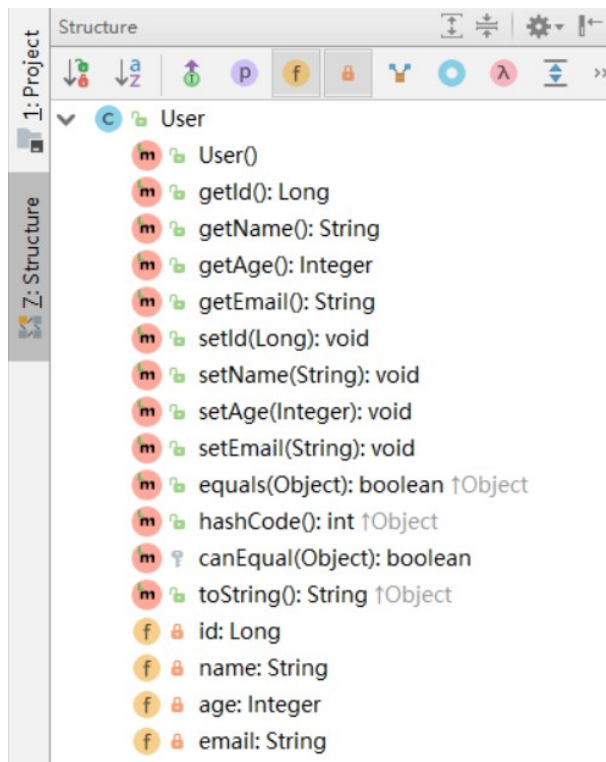
```
5 spring.datasource.password=123456
```

5、创建实体类

创建包 entity，编写实体类 User.java，并使用lombok简化实体类的编写

```
1 package com.atguigu.mybatisplus.entity;
2
3 @Data
4 public class User {
5     private Long id;
6     private String name;
7     private Integer age;
8     private String email;
9 }
```

查看编译结果



6、创建mapper接口

创建包 `mapper`，编写Mapper 接口： `UserMapper.java`

```
1 package com.atguigu.mybatisplus.mapper;
2
3 public interface UserMapper extends BaseMapper<User> {
4
5 }
```

7、启动类添加注解

在 Spring Boot 启动类中添加 `@MapperScan` 注解，扫描 Mapper 文件夹

```
1 package com.atguigu.mybatisplus;
2
3 @SpringBootApplication
4 @MapperScan("com.atguigu.mybatisplus.mapper")
5 public class MybatisPlusApplication {
6     .....
7 }
```

8、测试

添加测试类，进行功能测试：

```
1 package com.atguigu.mybatisplus;
2
3 @SpringBootTest
4 class MybatisPlusApplicationTests {
5
6     // @Autowired //默认按类型装配。是spring的注解
7     @Resource //默认按名称装配，找不到与名称匹配的bean，则按照类型装配。是J2EE的注
8     private UserMapper userMapper;
9
10    @Test
11    void testSelectList() {
```

```
12      //selectList()方法的参数: 封装了查询条件
13      //null: 无任何查询条件
14      List<User> users = userMapper.selectList(null);
15      users.forEach(System.out::println);
16  }
17 }
```

通过以上几个简单的步骤，我们就实现了 User 表的 CRUD 功能，甚至连 XML 文件都不用编写！

补充、查看sql输出日志

```
1 #mybatis日志
2 mybatis-plus.configuration.log-impl=org.apache.ibatis.logging.stdout.StdOutIn
```