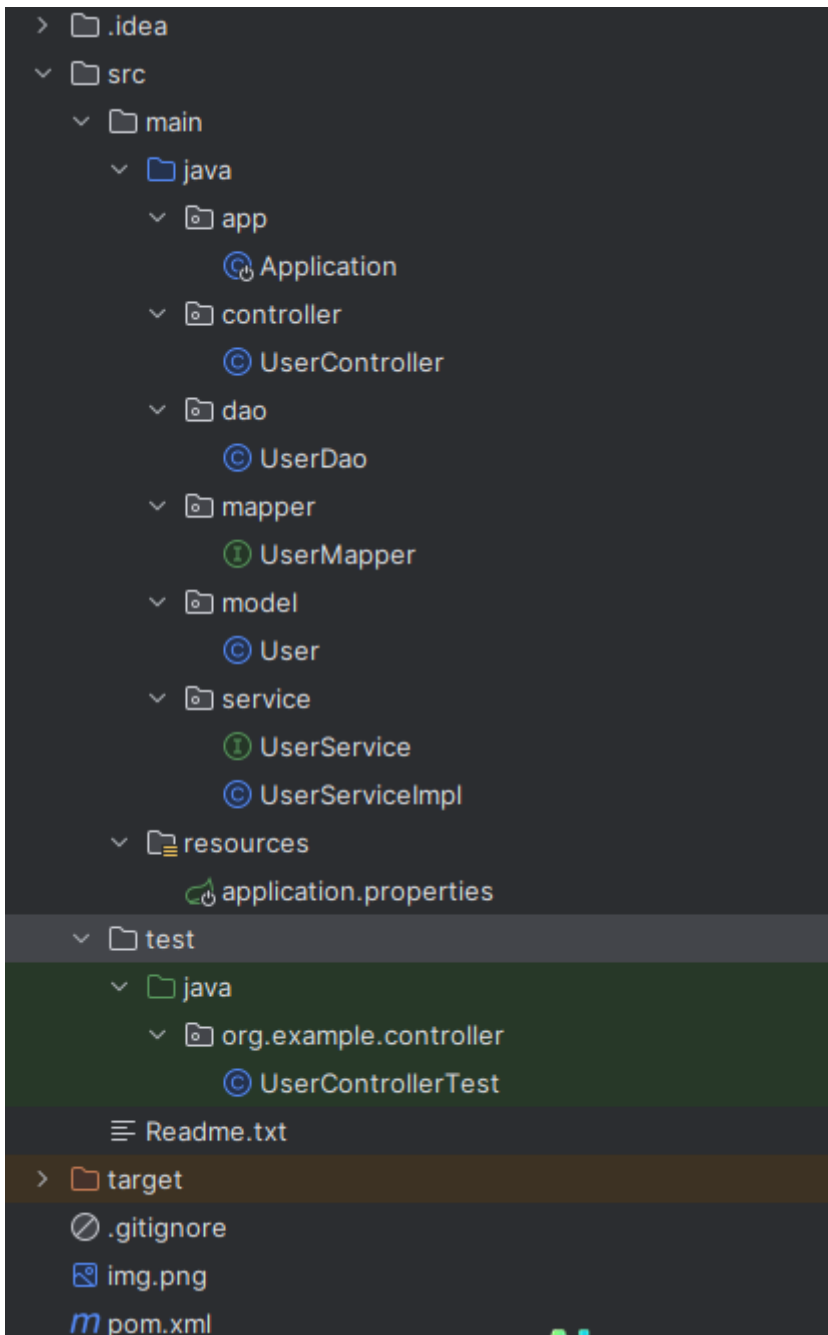


Springboot

简介

这是一个使用 Spring Boot、MyBatis 和 MySQL 实现的简单用户管理系统。项目包含用户信息的增、删、改、查的功能.

demo



DemoApplication.java class

这是包含'main'方法的Spring boot应用主类，通过调用 SpringApplication.run 方法启动 Spring Boot 应用。
@SpringBootApplication 注释用于启用 Spring Boot 的自动配置、组件扫描和 Spring Bean 的初始化。

UserController.java class

用户的控制器类，定义了用户增加、删除、update、获取的方法，并通过调用 UserService 来执行相应的操作。

UserService.java class

用户服务接口，定义了用户相关的业务操作。这些方法将由 UserServiceImpl 实现。

UserServiceimpl.java class

用户服务实现类，实现了 UserService 接口。通过调用 UserDao 执行数据库操作，实现了用户的增删改查功能

UserDao.java class

用户数据访问对象，负责与数据库交互。通过调用 UserMapper 中定义的 MyBatis 注释，实现用户信息的数据库操作

UserMapper.java class

用户映射器接口，使用 MyBatis 注释定义 SQL 操作。每个方法都对应一个 SQL 语句，用于插入、删除、更新和查询用户信息

User.java class

用户模型类，定义了用户的属性（如 ID、用户名、性别、年龄等）以及相应的 getter 和 setter 等基础方法

application.properties class

Spring Boot 应用的配置文件，包含数据库连接配置和 MyBatis 的配置

pom.xml

Maven 项目的配置文件，定义了项目的依赖项和构建插件。包含 Spring Boot 的依赖项、MyBatis 的依赖项、MySQL 的连接器、JUnit 5 和 Mockito 的测试依赖项

数据库创建（已在本地编译器中导入sql文件）

```
CREATE DATABASE db1;
```

```
USE db1;
```

```
CREATE TABLE users ( id BIGINT AUTO_INCREMENT PRIMARY KEY COMMENT '主键',  
  
username VARCHAR(50) NOT NULL COMMENT '用户名', gender TINYINT(1) NOT NULL COMMENT '性别',  
age INT NOT NULL COMMENT '年龄', create_user VARCHAR(40) NOT NULL COMMENT '创建人标识',  
create_user_name VARCHAR(128) NOT NULL COMMENT '创建人姓名', create_time DATETIME NOT NULL  
COMMENT '创建时间', update_user VARCHAR(40) NOT NULL COMMENT '最后更新人标识',  
update_user_name VARCHAR(128) NOT NULL COMMENT '最后更新人姓名', update_time DATETIME NOT  
NULL COMMENT '最后更新时间', is_delete TINYINT(1) DEFAULT 0 NOT NULL COMMENT '删除标记：0未删  
除 1已删除' );
```