# 第4章：springboot整合mybatis

## 创建项目

### 在pom.xml文件添加依赖

|  |
| --- |
| <dependencies>  <!-- thymeleaf依赖 -->  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-thymeleaf</artifactId>  </dependency>  <!-- springboot启动器 -->  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-web</artifactId>  </dependency>  <!-- 数据库驱动，默认最新版本 -->  <dependency>  <groupId>mysql</groupId>  <artifactId>mysql-connector-java</artifactId>  <scope>runtime</scope>  </dependency>  <!-- 内置tomcat -->  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-tomcat</artifactId>  <scope>provided</scope>  </dependency>  <!-- springboot测试启动器 -->  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-test</artifactId>  <scope>test</scope>  </dependency>  <!-- druid 数据库连接池 -->  <dependency>  <groupId>com.alibaba</groupId>  <artifactId>druid</artifactId>  <version>1.0.9</version>  </dependency>  <!-- mybatis启动器 -->  <dependency>  <groupId>org.mybatis.spring.boot</groupId>  <artifactId>mybatis-spring-boot-starter</artifactId>  <version>1.1.1</version>  </dependency>  </dependencies> |

### 编写application.properties全局配置文件

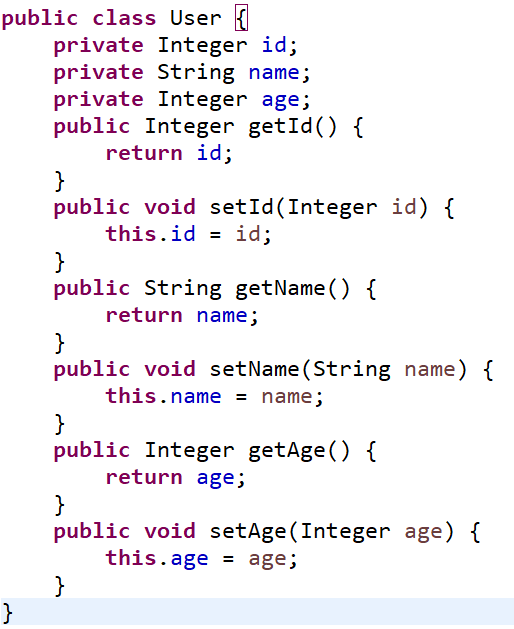
|  |
| --- |
| #加载驱动  spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver  #数据库连接路径  spring.datasource.url=jdbc:mysql://localhost:3306/test?useUnicode=true&characterEncoding=UTF-8&serverTimezone=UTC  #数据库用户名  spring.datasource.username=root  #数据库密码  spring.datasource.password=root  #数据源类型（阿里巴巴）  spring.datasource.type=com.alibaba.druid.pool.DruidDataSource  #mybatis取别名  mybatis.type-aliases-package=com.kazu.entity |

### 数据库设计

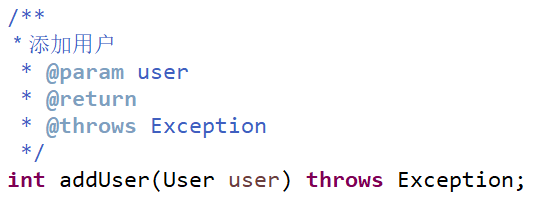
|  |
| --- |
| CREATE TABLE `users` (  `id` int(11) NOT NULL AUTO\_INCREMENT,  `name` varchar(255) DEFAULT NULL,  `age` int(11) DEFAULT NULL,  PRIMARY KEY (`id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8; |

## 添加用户

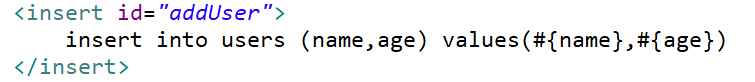
### 创建实体类



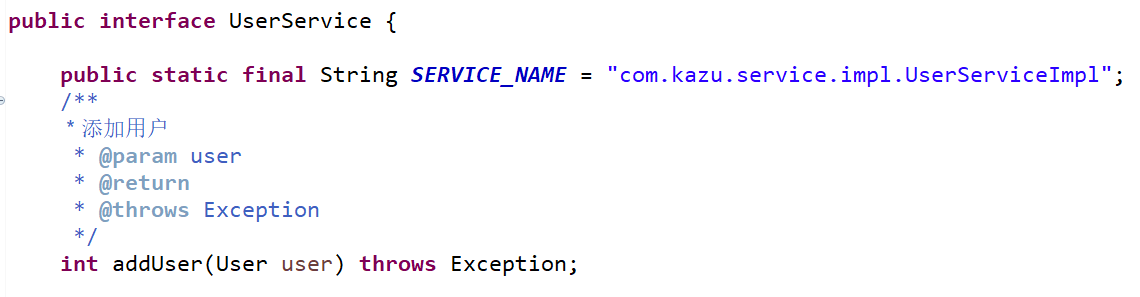
### 创建mapper接口



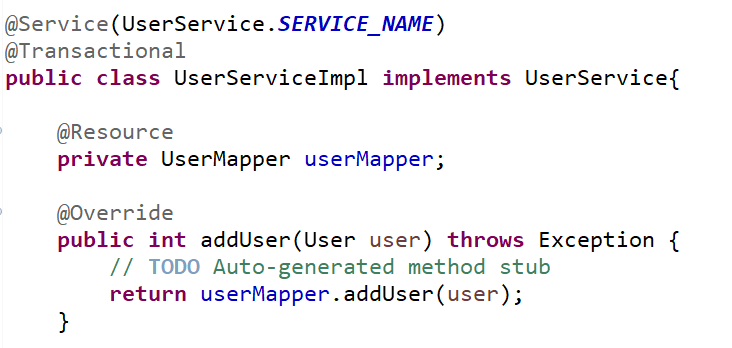
### 映射文件



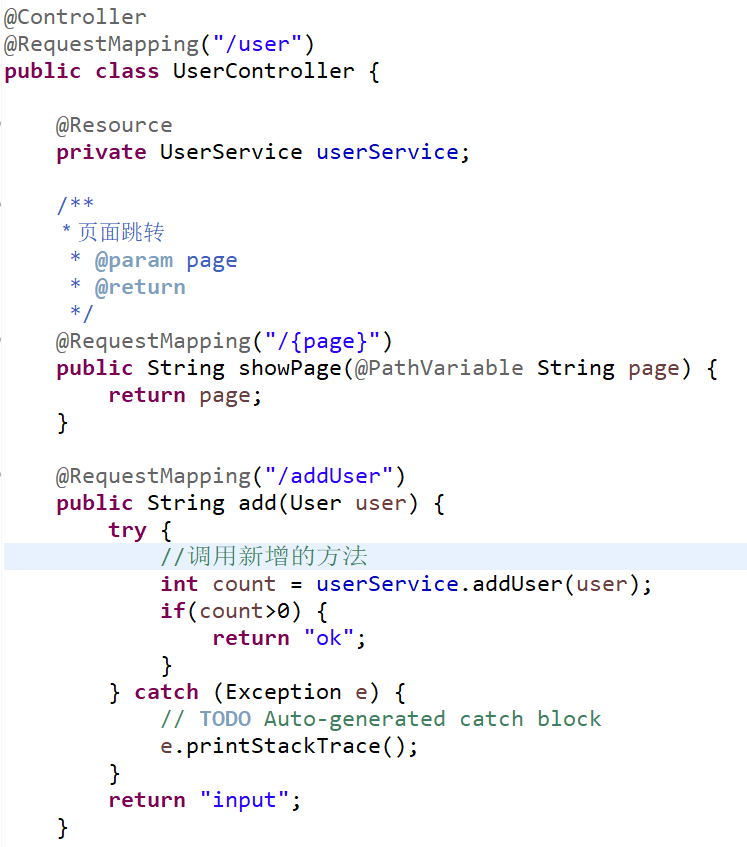
### 创建业务层



### 业务层实现类



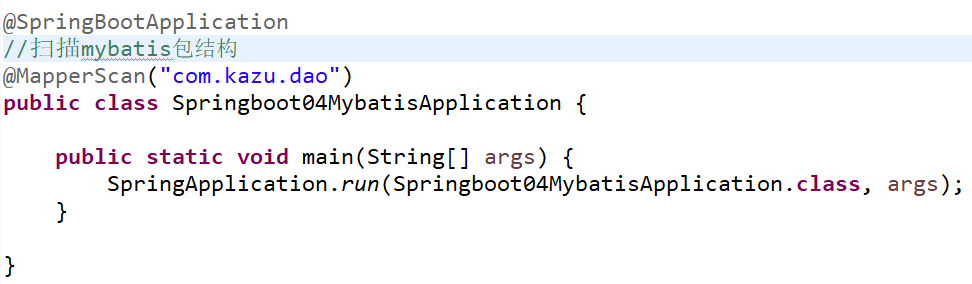
### 创建controller



### 编写页面

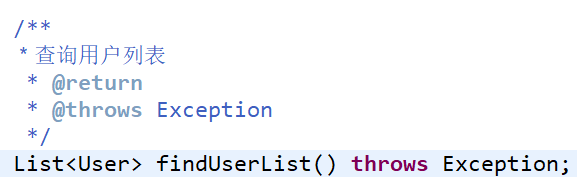


### 编写启动类

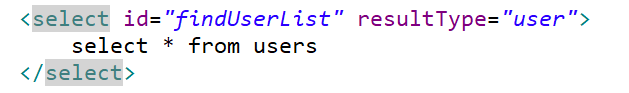


## 查询用户

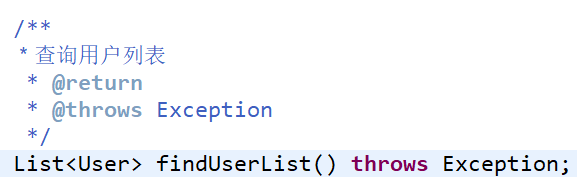
### 创建mapper接口



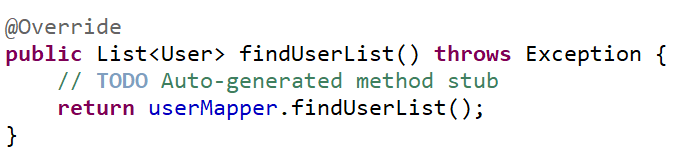
### 映射文件



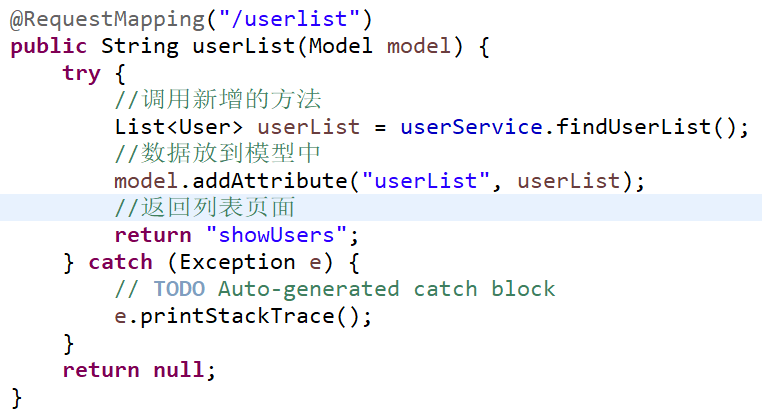
### 创建业务层



### 创建业务层实现类



### 创建controller

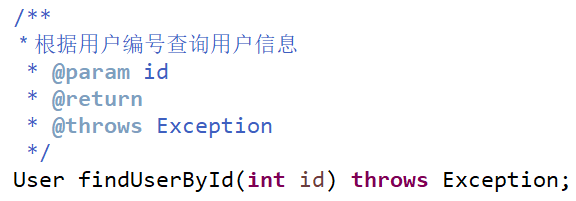


### 编写页面

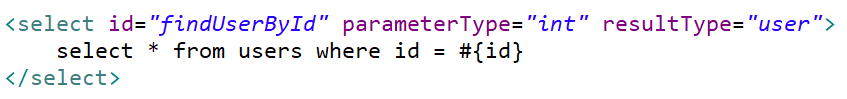


## 去到更新用户页面

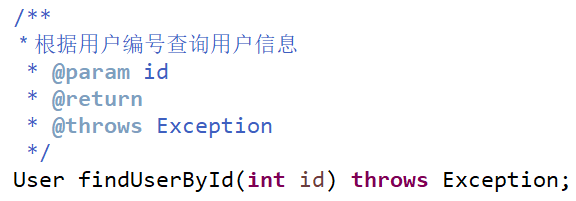
### 创建mapper接口



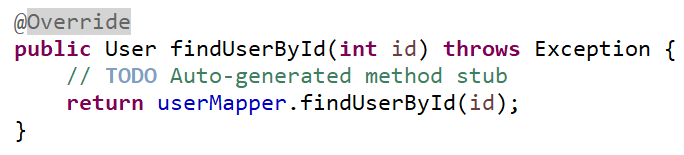
### 映射文件



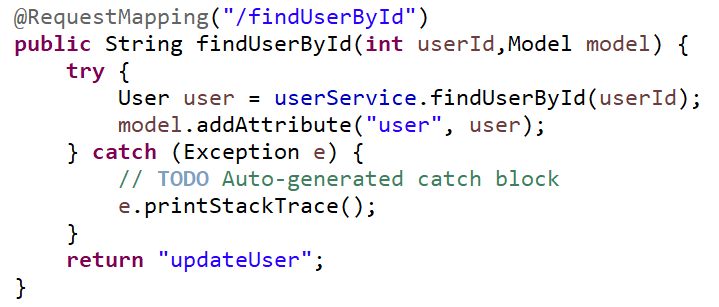
### 创建业务层



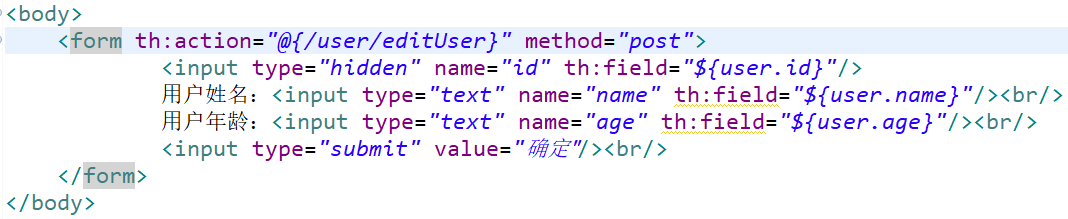
### 业务层实现类



### 创建controller

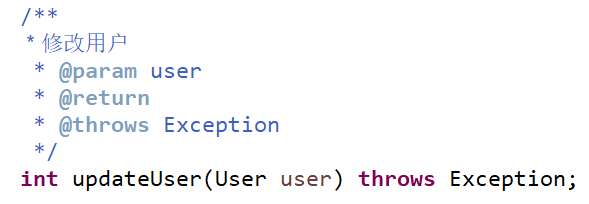


### 编写页面

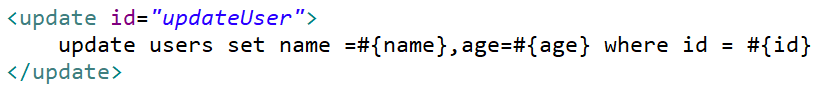


## 更新用户

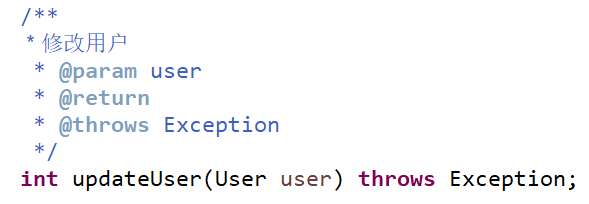
### 创建mapper接口



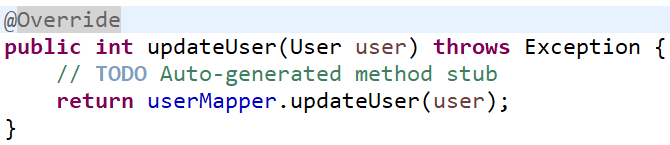
### 映射文件



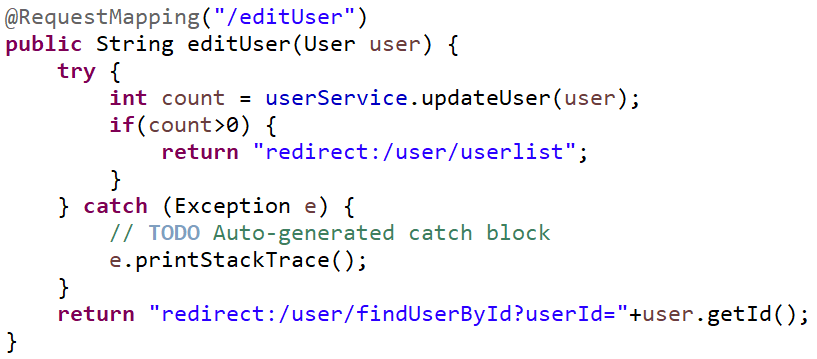
### 创建业务层



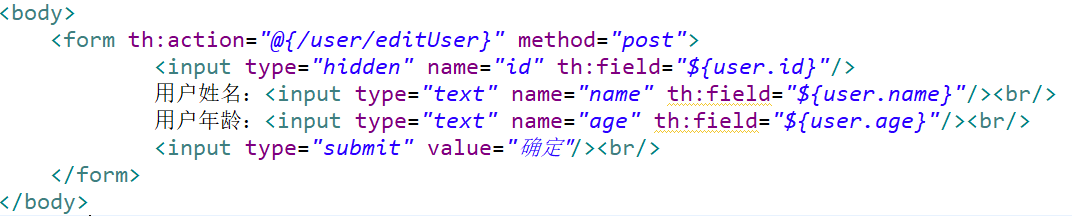
### 创建业务层实现类



### 创建controller

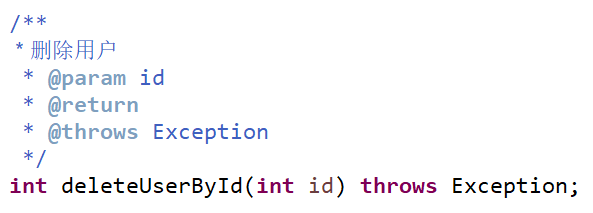


### 编写页面

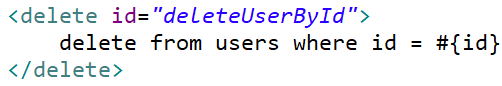


## 删除用户

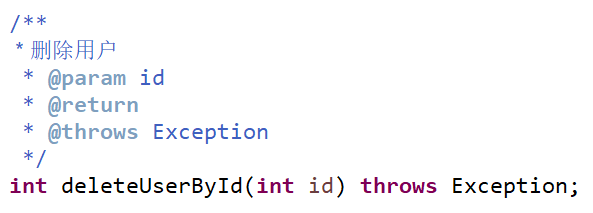
### 创建mapper接口



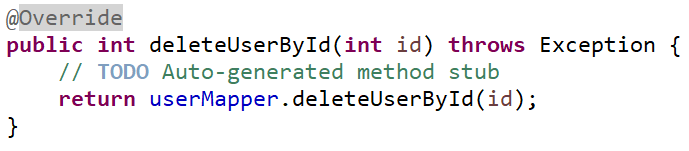
### 映射文件



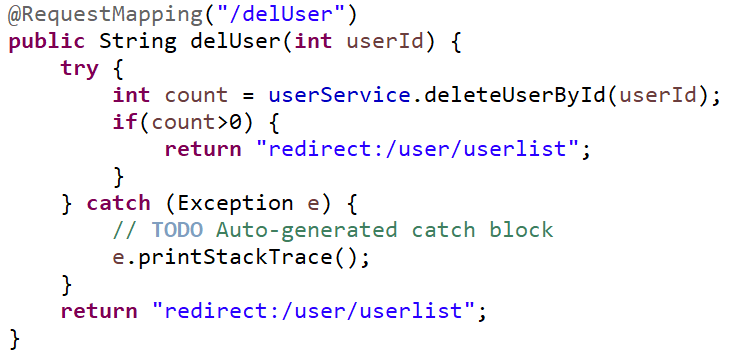
### 创建业务层



### 创建业务层实现类



### 创建controller



### 编写页面

