# 第4章：shiro集成springboot

## 阶段1：未集成下环境下

### 创建页面

#### webapp下添加index.html页面

|  |
| --- |
| <a href=*"users/add"*>用户添加</a><br>  <a href=*"users/update"*>用户修改</a><br> |

#### templates/user下添加add.html页面(内容为用户添加)

#### templates/user下添加update.html页面(内容为用户修改)

#### templates下添加login.html页面

|  |
| --- |
| <h2>登录</h2>  <form method=*"post"* action=*"users/login"*>  用户名:<input type=*"text"* name=*"userName"*/><br/>  密码：<input type=*"password"* name=*"password"*/><br/>  <input type=*"submit"* value=*"登录"*/>  </form> |

### UserController控制器

|  |
| --- |
|  |

添加以上代码，即可到浏览器测试，在index.html页面中测试

## 阶段2：springboot集成shiro实现登录验证

### springboot整合shiro

#### 在pom.xml文件中添加依赖

|  |
| --- |
| <!-- 导入thymeleaf依赖 -->  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-thymeleaf</artifactId>  </dependency>  <!-- shiro与spring整合依赖 -->  <dependency>  <groupId>org.apache.shiro</groupId>  <artifactId>shiro-spring</artifactId>  <version>1.4.0</version>  </dependency> |

#### 自定义Realm类

|  |
| --- |
|  |

#### 编写shiro配置类（核心）

|  |
| --- |
|  |

#### 使用shiro内置过滤器实现页面拦截

|  |
| --- |
|  |

#### UserController控制器

|  |
| --- |
| //去到登录页面  @RequestMapping("/tologin")  **public** String tologin() {  **return** "/login";  } |

### 整合MyBatis实现用户验证操作（登录）

#### 添加相关依赖

|  |
| --- |
| <!-- 添加热部署 -->  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-devtools</artifactId>  <optional>true</optional>  </dependency>  <!-- 导入mybatis相关的依赖 -->  <!-- 数据源 -->  <dependency>  <groupId>com.alibaba</groupId>  <artifactId>druid</artifactId>  <version>1.0.9</version>  </dependency>  <!-- mysql -->  <dependency>  <groupId>mysql</groupId>  <artifactId>mysql-connector-java</artifactId>  </dependency>  <!-- SpringBoot的Mybatis启动器 -->  <dependency>  <groupId>org.mybatis.spring.boot</groupId>  <artifactId>mybatis-spring-boot-starter</artifactId>  <version>1.1.1</version>  </dependency> |

#### 配置application.properties

|  |
| --- |
| #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/db\_shiro?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.shiro.entity |

#### 编写User实体类

|  |
| --- |
| **public** **class** User {  **private** Integer id;  **private** String userName;  **private** String password;    **public** Integer getId() {  **return** id;  }  **public** **void** setId(Integer id) {  **this**.id = id;  }  **public** String getUserName() {  **return** userName;  }  **public** **void** setUserName(String userName) {  **this**.userName = userName;  }  **public** String getPassword() {  **return** password;  }  **public** **void** setPassword(String password) {  **this**.password = password;  }  } |

#### 编写UserMapper接口

|  |
| --- |
| /\*\*  \* 根据用户名查询用户信息  \* **@param** userName  \* **@return**  \*/  **public** User getByUserName(String userName); |

#### 编写UserMapper映射文件

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"* ?>  <!DOCTYPE mapper  PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"  "http://mybatis.org/dtd/mybatis-3-mapper.dtd">  <mapper namespace=*"com.shiro.dao.UserMapper"*>  <resultMap type=*"User"* id=*"UserResult"*>  <result property=*"id"* column=*"id"*/>  <result property=*"userName"* column=*"userName"*/>  <result property=*"password"* column=*"password"*/>  </resultMap>    <select id=*"getByUserName"* parameterType=*"String"* resultMap=*"UserResult"*>  select \* from t\_user where userName=#{userName}  </select>  </mapper> |

#### 编写业务接口

|  |
| --- |
| /\*\*  \* 根据用户名查询用户信息  \* **@param** userName  \* **@return**  \*/  **public** User getByUserName(String userName); |

#### 编写业务实现

|  |
| --- |
|  |

#### 启动器添加@MapperScan注解

|  |
| --- |
|  |

#### 编写UserController控制器

|  |
| --- |
|  |

#### 编写登录页面

|  |
| --- |
|  |

#### 修改UserRealm

|  |
| --- |
|  |

### 整合MyBatis实现授权

#### 编写UserMapper接口

|  |
| --- |
|  |

#### 编写UserMapper映射文件

|  |
| --- |
|  |

#### 编写业务接口

|  |
| --- |
|  |

#### 编写业务实现类

|  |
| --- |
|  |

#### 修改realm

|  |
| --- |
|  |

#### 修改ShiroConfig配置类

|  |
| --- |
|  |

#### 无权访问控制器

|  |
| --- |
|  |