## 后台登录

* **登录页面设计+登录首页设计+后台管理模板提取**
* **登录页面的js控制用户名和密码非空**
* **UserService进行用户名和密码验证，从数据库中获取**

**//首先建立仓库，继承jpa仓库获取，jpa提供了一系列增删改查方法，但没有提供查询用户名密码的，所以在接口中自定义一个，jpa会自动查找**

**public interface** UserRepository **extends** JpaRepository<User,Long> {  
 User findByUsernameAndPassword(String username,String password);  
}

**//业务类的接口**

**public interface** UserService {  
 User checkUser(String username,String password);  
}

**//业务类的实现UserServiceImpl中的方法，调用dao包中的User仓库**

**public** User checkUser(String username, String password) {  
 User user = **userRepository**.findByUsernameAndPassword(username, DigestUtils.*md5DigestAsHex*(password.getBytes()));  
 **return** user;  
}

* **LoginController控制用户登录与注销，用户名和密码错误，提示message**

**使用post方式登录，注意指定method**

@Controller  
@RequestMapping(value = **"/admin"**,method = {RequestMethod.***POST***})  
**public class** LoginController {}

**使用RedirectAttributes attributes来获取重定向前设置的参数“message”**

@GetMapping(**"/login"**)  
**public** String login(@RequestParam String username, @RequestParam String password,  
 HttpSession session, RedirectAttributes attributes){  
 User user = **userService**.checkUser(username,password);  
 **if** (user!=**null**){  
 user.setPassword(**null**);  
 session.setAttribute(**"user"**,user);  
 **return "admin/index"**;  
 }**else**{  
 attributes.addFlashAttribute(**"message"**,**"用户名或密码错误！"**);  
 **return "redirect:/admin"**;  
 }  
}

**前端login.html使用如下方式显示参数“message”**

<**div class="ui mini negative message" th:unless="${#strings.isEmpty(message)}" th:text="${message}"**>

* **MD5加密，使用spring自带的DigestUtils.*md5DigestAsHex*：**

DigestUtils.*md5DigestAsHex*(**"111111"**.getBytes())

**输出的密文放在数据库中，登录的密码也用md5进行加密，密文与密文进行比对登录**

* **登录控制，在未登录情况下，无法访问博客：**

**创建LoginIntercepter，继承HandlerInterceptorAdapter，重写preHandle方法判断登录状态，从request参数中获取到session，判断session中是否带有user属性，没有则为未登录，重定向到登录界面**

**创建WebConfig，实现WebMvcConfigurer接口，重写addInterceptors，在此方法中添加过滤的路径，排除无需过滤的路径，一定要加上**@Configuration**注解。**