珞珈互助

一、 项目需求

运用多种设计模式，实现校园内部包括雨天需要送伞、小电炉接送、帮忙打印传送文件等互助服务，满足学生基本需要。

二、 项目实现

使用 Spring Boot + Spring MVC + MyBatis 为后端 layUI 为前端实现一个 web 网页，功能实现包括登录、发布需求、修改需求、接收需求等功能，并使用了登陆拦截器使 API 的调用更加安全

DAO 层

处理数据库

Service 层

处理数据后调用 DAO 层对数据库进行增删改查

登录部分

登录部分较为简单，只需要调用 service 层对提交数据进行验证即可，并且对拦截器方面注入验证，确保将未登录状态转为登录状态，如图

```java
@Controller
@RequestMapping("/login")
public class loginController {

    @Autowired
    UserService userService = null;

    @GetMapping
    public String loginPageGet() { return "login"; }

    @PostMapping
    public String loginProcess(@RequestParam("username") String name,
                               @RequestParam("password") String password,
                               Model model,
                               HttpSession session){
        if(password.equals(userService.getPasswordByUsername(name))){
            session.setAttribute( name: "userid",userService.getIdByUsername(name));
            session.setAttribute( name: "username",name);
            return "redirect:/home";
        }
        else{
            model.addAttribute( attributeName: "msg", attributeValue: "用户名或者密码错误");
            return "login";
        }
    }
}
```

Tips:向 session 中添加两个属性是为了登录后的请求需要用到这两个参数。

发布需求

发布需求通过对发布页面发送 POST 请求，随后对 POST 请求调用 service 层进行处理，如图：

```java
@GetMapping("/addOrder")
public String addOrder(Model model){
    model.addAttribute( attributeName: "order",new Order());
    return "addOrder";
}

@PostMapping("/addOrder")
public String insertOrder(@ModelAttribute(value="order") Order order,
                          HttpServletRequest request){
    order.setBelong((Long)request.getSession().getAttribute( name: "userid"));
    order.setCreatTime(new Date());
    orderService.addOrder(order);
    System.out.println(order);
    return "redirect:/home";
}
```

修改需求

修改需求首先需要验证权限，只有当该需求属于当前登录用户以及该需求未被接单时，才能对其进行修改（权限处理在详细页处理），当允许进行修改时，跳转到该订单的修改视图进行修改后提交 POST 请求即可，如图

```java
@GetMapping("/fixOrder/{orderId}")
public String fixOrder(@PathVariable(name ="orderId")Long orderId,Model model){
    model.addAttribute( attributeName: "order",orderService.displayOrderByOrderId(orderId));
    return "fixOrder";
}

@PostMapping("/fixOrder/{orderId}")
public String fixOrder(@PathVariable(name ="orderId")Long orderId,@ModelAttribute(value="order") Order order,Htt
    order.setBelong((Long)request.getSession().getAttribute( name: "userid"));
    order.setId(orderId);
    order.setCreatTime(new Date());
    orderService.updateOrder(order);
    return "redirect:/home";
}
```

需求详情

首先需要通过 GET 方式获取到该需求的详细情况并且验证权限，包括是否允许修改以及是否允许接单，权限验证在视图层，如图

视图层

```
<a th:if="${oneOrder.getBelong() ne nowUserId && flag eq 0}" class="layui-btn layui-btn-normal layui-btn-radius" th:href="{
<a th:if="${oneOrder.getBelong()} eq ${nowUserId}" class="layui-btn layui-btn-danger layui-btn-radius" th:href="@{'/home/f
<button class="layui-btn layui-btn-warm layui-btn-radius" onClick="history.go(-1)">返回</button>
```

控制层

```java
@GetMapping("/detail/{id}")
public String showDetail(@PathVariable(name="id") Long id, Model model,HttpServletRequest
    System.out.println(webRecord.getPageFlow());
    webRecord.comeDetailPage(id);
    Order order = orderService.displayOrderByOrderId(id);
    User user = userService.getUser(order.getBelong());
    model.addAttribute( attributeName: "owner",user.getUsername());
    model.addAttribute( attributeName: "oneOrder",order);
    model.addAttribute( attributeName: "nowUserId",request.getSession().getAttribute( name: "
    int flag=0;
    List<Long> orders = orderService.getAllAcceptOrders();
    if(orders.contains(id)){
        user =userService.getUser(orderService.getOrderAccepter(id));
        model.addAttribute( attributeName: "accepter",user.getUsername());
        flag = 1;
    }
    model.addAttribute( attributeName: "flag",flag);
    return "detail";
}
```

接收需求

只需要发送一个 GET 请求即可，控制层进行处理后调用 Service 层，如图

```java
@GetMapping("/acceptOrder/{orderId}")
public String acceptOrder(@PathVariable(name="orderId") Long orderId,HttpServletRequest request){
    Order order = orderService.displayOrderByOrderId(orderId);
    orderService.userAccepted(order.getId(),(Long)request.getSession().getAttribute( name: "userid"),order.getBelong());
    return "redirect:/home";
}
```

获取所有订单、自己发布订单、已接受订单

三个 GET 请求并返回 JSON 串用来填充 LayUI 的 table 组件，如图

```java
@GetMapping("/accept")
@ResponseBody
public Map<String, Object> getAccept(HttpServletRequest request){
    Map<String,Object> map = new HashMap<String,Object>();
    map.put("code",0);
    map.put("msg","");
    map.put("data",orderService.getAcceptOrder((Long)request.getSession().getAttribute( name: "userid")));
    return map;
}

@GetMapping("/release")
@ResponseBody
public Map<String,Object> getRelease(HttpServletRequest request){
    Map<String,Object> map = new HashMap<~>();
    map.put("code",0);
    map.put("msg","");
    map.put("data",orderService.getAllSelfOrder((Long)request.getSession().getAttribute( name: "userid")));
    return map;
}

@GetMapping("/orders")
@ResponseBody
public Map<String,Object> getAllOrders() {
    Map<String,Object> map = new HashMap<~>();
    map.put("code",0);
    map.put("msg","");
    map.put("data",orderService.getAllOrder());
    return map;
}
```

三、 功能展示

项目内附件视频 WhuHelper.mp4 用以显示项目展示。