乐字节教育高级架构课程

正所谓“授人以鱼不如授人以渔”，你们想要的 Java 学习资料来啦！不管你是学生，还是已经步入职场的同行，希望你们都要珍惜眼前的学习机会，奋斗没有终点，知识永不过时。

扫描下方二维码即可领取

乐字节晓啡

乐字节官方交流群

订单系统搜索功能和安全退出功能实现

订单系统编写全局拦截器

shop-order的OrderCommonInterceptor.java

```
package com.xxxx.order.interceptor;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Component;
import org.springframework.util.StringUtils;
import org.springframework.web.servlet.HandlerInterceptor;
import org.springframework.web.servlet.ModelAndView;

import javax.servlet.ServletContext;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * 全局拦截器
 *
 * @author zhoubin
 * @since 1.0.0
 */
@Component
public class OrderCommonInterceptor implements HandlerInterceptor {

   @Value("${shop.portal.url}")
   private String shopPortalUrl;

   //请求处理的方法之前执行
   //如果返回true，执行下一个拦截器或者目标程序，如果返回false，不执行下一个拦截器或者目标程序
   @Override
   public boolean preHandle(HttpServletRequest request, HttpServletResponse response,
Object handler) throws Exception {
      //获取Application对象中的url地址
      ServletContext context = request.getSession().getServletContext();
      String portalUrl = (String) context.getAttribute("portalUrl");
      if (StringUtils.isEmpty(portalUrl)){
         context.setAttribute("portalUrl",shopPortalUrl);
      }
      return true;
   }

   //请求处理的方法之后执行
   @Override
   public void postHandle(HttpServletRequest request, HttpServletResponse response, Object
handler, ModelAndView modelAndView) throws Exception {

   }
```

```
    //处理后执行清理工作
    @Override
    public void afterCompletion(HttpServletRequest request, HttpServletResponse response,
Object handler,
                                Exception ex) throws Exception {

    }
}
```

ego-order的MvcConfig.java

```
/**
 * addInterceptor:添加自定义拦截器
 * addPathPatterns: 添加拦截请求 /**表示拦截所有
 * excludePathPatterns: 不拦截的请求
 *
 * @param registry
 */
@Override
public void addInterceptors(InterceptorRegistry registry) {
    registry.addInterceptor(commonInterceptor)
            .addPathPatterns("/**");
    registry.addInterceptor(loginInterceptor)
            .addPathPatterns("/**")
            .excludePathPatterns("/static/**")
            .excludePathPatterns("/login/**")
            .excludePathPatterns("/image/**")
            .excludePathPatterns("/user/login/**")
            .excludePathPatterns("/user/logout/**");
}
```

订单系统添加Controller

shop-order的SearchController.java

```
package com.xxxx.order.controller;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;

import javax.servlet.http.HttpServletRequest;
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;

/**
 * 搜索Controller
 *
 * @author zhoubin
 * @since 1.0.0
 */
```

```
@Controller
@RequestMapping("search")
public class SearchController {

    /**
     * 跳转搜索页面
     * @param request
     * @param searchStr
     * @param model
     * @return
     */
    @RequestMapping("index")
    public String index(HttpServletRequest request, String searchStr, Model model) {
        try {
            // 对输入的内容进行编码, 防止中文乱码
            searchStr = URLEncoder.encode(searchStr, "UTF-8");
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        }
        return "redirect:"
                + request.getSession().getServletContext().getAttribute("portalUrl")
                + "search/index?searchStr=" + searchStr;
    }
}
```

shop-order的UserController.java

```
package com.xxxx.order.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;

import javax.servlet.http.HttpServletRequest;

/**
 * 用户Controller
 *
 * @author zhoubin
 * @since 1.0.0
 */
@Controller
@RequestMapping("user")
public class UserController {

    /**
     * 用户退出
     * @param request
     * @return
     */
    @RequestMapping("logout")
    public String logout(HttpServletRequest request){
        return "redirect:"
                + request.getSession().getServletContext().getAttribute("portalUrl")
                + "user/logout";
    }
}
```

订单系统页面处理

将前台系统的common/search.ftl和common/welcome.ftl两个页面直接拷贝过来即可。

请求会先到达订单系统对应的方法，然后重定向至前台系统。