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

前台系统编写拦截器

shop-portal的PortalCartInterceptor.java

```
package com.xxxx.portal.interceptor;

import com.alibaba.dubbo.config.annotation.Reference;
import com.xxxx.common.util.CookieUtil;
import com.xxxx.common.util.JsonUtil;
import com.xxxx.sso.pojo.Admin;
import com.xxxx.sso.service.SSOService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.data.redis.core.RedisTemplate;
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.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.util.concurrent.TimeUnit;

/**
 * 前台登录拦截器
 *
 * @author zhoubin
 * @since 1.0.0
 */
@Component
public class PortalLoginInterceptor implements HandlerInterceptor {

   @Reference(interfaceClass = SSOService.class)
   private SSOService ssoService;

   @Value("${user.ticket}")
   private String userTicket;

   private static final String COOKIE_NAME = "userTicket";

   @Autowired
   private RedisTemplate<String, String> redisTemplate;

   //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 {
      //获取用户名票据
      String ticket = CookieUtil.getCookieValue(request, COOKIE_NAME);
```

```
        if (!StringUtils.isEmpty(ticket)) {
            //如果票据存在，进行验证
            Admin admin = ssoService.validate(ticket);
            if (null!=admin){
                //将用户信息添加至session中，用于页面返显
                request.getSession().setAttribute("user", admin);
                //重新设置新的失效时间
                redisTemplate.opsForValue().set(userTicket + ":" + ticket,
JsonUtil.object2JsonStr(admin), 30,
                        TimeUnit.MINUTES);
                return true;
            }else {
                //清除session的信息
                request.getSession().removeAttribute("user");
            }
        }else {
            //清除session的信息
            request.getSession().removeAttribute("user");
        }
        //票据不存在或者用户验证失败，重定向至登录页面
        response.sendRedirect(request.getContextPath() + "/login");
        return false;
    }

    //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 {

    }
}
```

shop-portal的MvcConfig.java

```
package com.xxxx.portal.config;

import com.xxxx.portal.interceptor.PortalLoginInterceptor;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Configuration;
import org.springframework.web.servlet.config.annotation.EnableWebMvc;
import org.springframework.web.servlet.config.annotation.InterceptorRegistry;
import org.springframework.web.servlet.config.annotation.ResourceHandlerRegistry;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;
```

如果需要更多优质的Java、Python、架构、大数据等IT资料请加微信：lezijie007

```java
/**
 * MVC配置类
 *
 * @author zhoubin
 * @since 1.0.0
 */
@Configuration
@EnableWebMvc
public class MvcConfig implements WebMvcConfigurer {

   @Autowired
   private PortalLoginInterceptor loginInterceptor;

   /**
    * 放行静态资源
    *
    * @param registry
    */
   @Override
   public void addResourceHandlers(ResourceHandlerRegistry registry) {
      registry.addResourceHandler("/static/**").addResourceLocations("classpath:/static/");
   }

   /**
    *  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(loginInterceptor)
            .addPathPatterns("/cart/**")
            .excludePathPatterns("/static/**")
            .excludePathPatterns("/login/**")
            .excludePathPatterns("/user/login/**")
            .excludePathPatterns("/user/logout/**");
   }
}
```

前台系统编写Controller

shop-portal的CartController.java

```java
package com.xxxx.portal.controller;

import com.alibaba.dubbo.config.annotation.Reference;
import com.xxxx.common.pojo.Admin;
import com.xxxx.common.result.BaseResult;
import com.xxxx.rpc.service.CartService;
import com.xxxx.rpc.vo.CartResult;
import com.xxxx.rpc.vo.CartVo;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;

import javax.servlet.http.HttpServletRequest;
import java.util.Date;

/**
 * 购物车Controller
 *
 * @author zhoubin
 * @since 1.0.0
 */
@Controller
@RequestMapping("cart")
public class CartController {
   @Reference(interfaceClass = CartService.class)
   private CartService cartService;

   /**
    * 加入购物车
    *
    * @param cart
    * @param request
    * @return
    */
   @RequestMapping("addCart")
   @ResponseBody
   public BaseResult addCart(CartVo cart, HttpServletRequest request) {
      cart.setAddTime(new Date());
      Admin admin = (Admin) request.getSession().getAttribute("user");
      return cartService.addCart(cart, admin);
   }

   /**
    * 获取购物车数量
    *
    * @param request
    * @return
    */
   @RequestMapping("getCartNum")
   @ResponseBody
   public Integer getCartNum(HttpServletRequest request) {
      Admin admin = (Admin) request.getSession().getAttribute("user");
      return cartService.getCartNum(admin);
   }

}
```

前台系统提取购物车信息公共页面

shop-portal的common/cart.ftl

如果需要更多优质的Java、Python、架构、大数据等IT资料请加微信：lezijie007

```
<!-- 设置项目根路径全局变量 -->
<#assign ctx=request.contextPath/>
<input type="hidden" id="userCartHasUserName" value="${(user.userName)!''}"/>
<div id="s_cart">
    <ul>
        <li class="nums">
            <a href="">购物车: <span id="s_cart_nums1">0</span> 件</a>
            <a href="" class="btn" id="s_cart_nums2"></a>
        </li>
        <li class="checkout">
            <a href="#">去结算&gt;&gt;</a>
        </li>
    </ul>
</div>
<script type="text/javascript">
    $(function () {
        // 获取购物车数量
        if ($("#userCartHasUserName").val())
            getCartNum();
    });

    // 添加至购物车
    function addToCart(goodsId, goodsName, marketPrice, originalImg, goodsNum=1) {
        $.ajax({
            url: "${ctx}/cart/addCart",
            type: "POST",
            data: {
                goodsId: goodsId,
                goodsName: goodsName,
                marketPrice: marketPrice,
                originalImg: originalImg,
                goodsNum: goodsNum
            },
            dataType: "JSON",
            success: function (result) {
                if (200 == result.code) {
                    var num = parseInt($("#s_cart_nums1").text());
                    $("#s_cart_nums1").text(num + 1);
                    layer.msg("添加至购物车成功");
                }
            },
            error: function (result) {
                console.log(result);
            // status=200 statusText="OK"说明请求正常, 无返回结果
                if (200 == result.status) {
                    // 跳转登录页面
                    location.href = "${ctx}/login";
                } else {
                    // 系统真的出错了
                    layer.alert("亲, 系统正在升级中, 请稍后再试! ");
                }
            }
        });
    }

    // 获取购物车数量
    function getCartNum() {
        $.ajax({
            url: "${ctx}/cart/getCartNum",
            type: "POST",
            dataType: "JSON",
            success: function (result) {
                $("#s_cart_nums1").text(result);
            },
            error: function (result) {
                layer.alert("亲，系统正在升级中，请稍后再试！");
            }
        });
    }
</script>
```

修改pages/search/doSearch.ftl(以后所有前台系统会出现该信息的页面都要替换)，将原来该部分的代码替换为如下代码

```
<#include "../common/cart.ftl">
```

比如index.ftl页面替换为

```
<#include "common/cart.ftl">
```

乐字节 LEBYTE

如果需要更多优质的Java、Python、架构、大数据等IT资料请加微信：lezijie007