## 工作日志

学号：1830090115

姓名：李响

* 学习内容：

|  |  |
| --- | --- |
| 序号 | 内容 |
| 1 | 购物车的实现。 |
| 2 | sso系统集成portal。   * 1. 用户登录注册功能都跳转到sso系统。   2. 登录完成后跳转首页。   3. 首页需要在加载完成后，需要从cookie中取token，调用sso系统的服务（jsonp）根据token查询用户信息，把用户名显示到首页。   4. 使用拦截器实现用户强制登录。   5. 回调页面的实现 |
| 3 | 登录拦截器。 |

* 心得体会：

登录拦截器：

package top.lixiang007.taotao.sso.interceptor;  
  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.beans.factory.annotation.Value;  
import org.springframework.web.servlet.HandlerInterceptor;  
import org.springframework.web.servlet.ModelAndView;  
import top.lixiang007.taotao.pojo.TbUser;  
import top.lixiang007.taotao.sso.service.UserService115;  
  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
  
*/\*\*  
 \* 用户登录拦截器  
 \*/  
/\*\*  
 \** ***@Author*** *Xiang Li  
 \** ***@Date*** *2021/9/6 上午8:53  
 \** ***@Since*** *version-1.0  
 \** ***@青软实训*** *\*/*public class LoginInterceptor115 implements HandlerInterceptor {  
  
 @Autowired  
 private UserService115 userService;  
 @Value("${SSO\_LOGIN\_URL}")  
 private String SSO\_LOGIN\_URL;  
  
 @Override  
 public boolean preHandle(HttpServletRequest httpServletRequest, HttpServletResponse httpServletResponse, Object o) throws Exception {  
 TbUser user = userService.getUserByToken(httpServletRequest, httpServletResponse);  
 // 如果用户处于未登录状态，跳转到登录页面  
 if (user == null) {  
 httpServletResponse.sendRedirect(SSO\_LOGIN\_URL + "?redirectUrl=" + httpServletRequest.getRequestURL());  
 return false;  
 }  
 // 如果取到用户信息，放行  
 httpServletRequest.setAttribute("user", user);  
 return true;  
 }  
  
 @Override  
 public void postHandle(HttpServletRequest httpServletRequest, HttpServletResponse httpServletResponse, Object o, ModelAndView modelAndView) throws Exception {  
  
 }  
  
 @Override  
 public void afterCompletion(HttpServletRequest httpServletRequest, HttpServletResponse httpServletResponse, Object o, Exception e) throws Exception {  
  
 }  
}

**1、添加购物车不需要用户登录。购物车的数据应该放到cookie中。**

2、当向购物车添加同一款商品时，购物车中商品的数量增加。

3、购物车中可以删除商品

4、购物车中可以修改商品数量。商品的总价需要重新计算。

**5、点击“结算”按钮要求用户登录。**

分析

请求的url：/cart/add/${itemId}.html

1、接收商品id，根据商品id取商品的图片、title、单价，需要包含商品的数量。并且把数据保存到列表中。

2、把商品数据转换成json数据，保存到cookie中。对购物车数据进行转码。

3、展示添加购物车成功页面。

Service

参数：

1、商品id

2、商品数量

业务逻辑：

1. 接收商品id
2. 从cookie中购物车商品列表
3. 从商品列表中查询列表是否存在此商品
4. 如果存在商品的数量加上参数中的商品数量
5. 如果不存在，调用rest服务，根据商品id获得商品数据。
6. 把商品数据添加到列表中
7. 把购物车商品列表写入cookie
8. 返回TaotaoResult

返回值：TaotaoResult

购物车controller实现。

package top.lixiang007.taotao.portal.controller;  
  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.Model;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.ResponseBody;  
import top.lixiang007.taotao.common.pojo.TaotaoResult;  
import top.lixiang007.taotao.portal.pojo.CartItem;  
import top.lixiang007.taotao.portal.service.CartService;  
  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import java.util.List;  
  
*/\*\*  
 \* 购物车Controller  
 \*/  
/\*\*  
 \** ***@Author*** *Xiang Li  
 \** ***@Date*** *2021/9/6 上午9:35  
 \** ***@Since*** *version-1.0  
 \** ***@青软实训*** *\*/*@Controller  
@RequestMapping("/cart")  
public class CartController115 {  
  
 @Autowired(required = false)  
 private CartService cartService;  
  
 @RequestMapping("/add/{itemId}")  
 public String addCart(@PathVariable Long itemId, Integer num, HttpServletRequest request, HttpServletResponse response) {  
 TaotaoResult result = cartService.addCart(itemId, num, request, response);  
 return "cart-success";  
 }  
  
 @RequestMapping("/cart")  
 public String showCart(Model model, HttpServletRequest request) {  
 List<CartItem> cartItems = cartService.getCartItems(request);  
 model.addAttribute("cartList", cartItems);  
 return "cart";  
 }  
  
 @RequestMapping("/update/num/{itemId}/{num}")  
 @ResponseBody  
 public TaotaoResult updateCartItemNum(@PathVariable Long itemId, @PathVariable Integer num, HttpServletResponse response, HttpServletRequest request) {  
 return cartService.updateCartItem(itemId, num, request, response);  
 }  
  
 @RequestMapping("/delete/{itemId}")  
 public String deleteCartItem(@PathVariable Long itemId, HttpServletResponse response, HttpServletRequest request) {  
 TaotaoResult result = cartService.deleteCartItem(itemId, request, response);  
 return "redirect:/cart/cart.html";  
 }  
}

1、未登录的情况下，可以把购物车写入cookie。

2、已经登录的情况下，需要把购物车写入redis

3、登录时判断cookie中有购物车商品，应该吧cookie中的购物车商品列表转移到redis中。

Key:用户id

Value：购物车商品列表

1. 如果想redis中转移商品时，redis的购物车中已经有商品，此时，需要把商品合并。如果是同一款商品数量叠加，如果新商品就合并商品。

* 遇到问题：

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | <http://localhost:8084/user/logout.html>请求失败。 |

* 解决方案：

|  |  |
| --- | --- |
| 序号 | 方案 |
| 1 | Redis服务器没有设置密码，故客户端不发动密码，注释掉该行代码。  //jedis.auth(REDIS\_PASSWORD); |

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

图形用户界面, 应用程序

描述已自动生成