

[今日课程大纲]

购物车需求 实现购物车

[知识点详解]

- 一. 新建 ego-cart 并完成拦截器功能
- 1. 搭建 ego-cart
- 2. 在 ego-cart 中新建拦截器 com.ego.cart.interceptor.LoginInterceptor

```
public class LoginInterceptor implements
HandlerInterceptor {
    @Override
    public boolean preHandle(HttpServletRequest request,
HttpServletResponse response, Object handler)
        throws Exception {
        String token = CookieUtils.getCookieValue(request,
"TT_TOKEN");
        if(token!=null&&!token.equals("")){
            String json =
HttpClientUtil.doPost("http://localhost:8084/user/tok
en/"+token);
        EgoResult er = JsonUtils.jsonToPojo(json,
```

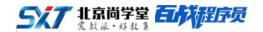


```
EgoResult.class);
    if(er.getStatus()==200){
        return true;
    }
    String num = request.getParameter("num");
    response.sendRedirect("http://localhost:8084/user/s
howLogin?interurl="+request.getRequestURL()+"%3Fnum="
+num);
    return false;
}
```

- 3. 在 ego-cart 的 springmvc.xml 中配置拦截器
 - 3.1 拦截器全部

4. 修改 ego-passport 的 TbUserController 的 showLogin 方法

```
@RequestMapping("user/showLogin")
```



```
public String
showLogin(@RequestHeader(value="Referer",defaultValue
="") String url,Model model,String interurl){
    if(interurl!=null&&!interurl.equals("")){
        model.addAttribute("redirect", interurl);
    }else if(!url.equals("")){
        model.addAttribute("redirect", url);
    }
    return "login";
}
```

二.添加购物车业务

1. 在 ego-commons 的 commons.properties 添加

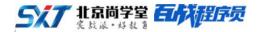
```
# ego-cart LoginIntercepor not use
passport.url = http://localhost:8084/user/token/
cart.key =cart:
```

2. 在 ego-cart 中新建 CartService 及实现类

```
public interface CartService {
```



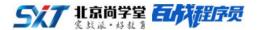
```
/**
   * 加入购物车
   * @param id
   * @param num
   */
  void addCart(long id,int num,HttpServletRequest
request);
}
@Service
public class CartServiceImpl implements CartService {
  @Resource
  private JedisDao jedisDaoImpl;
  @Reference
  private TbItemDubboService tbItemDubboServiceImpl;
  @Value("${passport.url}")
  private String passportUrl;
  @Value("${cart.key}")
  private String cartKey;
  @Override
  public void addCart(long id, int
num,HttpServletRequest request) {
     //集合中存放所有购物车商品
```



```
List<TbItemChild> list = new ArrayList<>();
     //redis 中的 key
     String token = CookieUtils.getCookieValue(request,
"TT TOKEN");
     String jsonUser =
HttpClientUtil.doPost(passportUrl+token);
     EgoResult er = JsonUtils.jsonToPojo(jsonUser,
EgoResult.class);
     String key =
cartKey+((LinkedHashMap)er.getData()).get("username")
     //如果 redis 中存在 key
     if(jedisDaoImpl.exists(key)){
        String json = jedisDaoImpl.get(key);
        if(json!=null&&!json.equals("")){
           list = JsonUtils.jsonToList(json,
TbItemChild.class);
          for (TbItemChild tbItemChild : list) {
             if((long)tbItemChild.getId()==id){
                //购物车中存在该商品
```



```
tbItemChild.setNum(tbItemChild.getNum()+num);
                jedisDaoImpl.set(key,
JsonUtils.objectToJson(list));
                return ;
              }
           }
           //重新添加到 redis 中
        }
     }
     TbItem item = tbItemDubboServiceImpl.selById(id);
     TbItemChild child = new TbItemChild();
     child.setId(item.getId());
     child.setTitle(item.getTitle());
  child.setImages(item.getImage()==null||item.getImag
e().equals("")?new
String[1]:item.getImage().split(","));
     child.setNum(num);
     child.setPrice(item.getPrice());
     list.add(child);
```

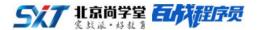


```
jedisDaoImpl.set(key,

JsonUtils.objectToJson(list));
}
```

3. 在 ego-cart 中 CartController 修改如下

```
@Controller
public class CartController {
  @Resource
  private CartService cartServiceImpl;
  /**
   * 添加购物车
   * @param id
   * @return
   */
  @RequestMapping("cart/add/{id}.html")
  public String addCart(@PathVariable long id,int
num,HttpServletRequest request){
     cartServiceImpl.addCart(id, num, request);
     return "cartSuccess";
```

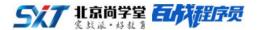


```
}
```

三.显示购物车信息

1. 在 ego-cart 的 CartService 添加

```
/**
   * 显示购物车
   * @return
   */
                           showCart(HttpServletRequest
  List<TbItemChild>
request);
@Override
  public List<TbItemChild> showCart(HttpServletRequest
request) {
     //redis 中的 key
     String token = CookieUtils.getCookieValue(request,
"TT_TOKEN");
     String jsonUser =
HttpClientUtil.doPost(passportUrl+token);
     EgoResult er = JsonUtils.jsonToPojo(jsonUser,
```



```
EgoResult.class);
    String key =
cartKey+((LinkedHashMap)er.getData()).get("username")
;

String json = jedisDaoImpl.get(key);
    return JsonUtils.jsonToList(json,
TbItemChild.class);
}
```

2. 在 ego-cart 的 CartController 添加

```
/**

* 显示购物车

* @return

*/

@RequestMapping("cart/cart.html")

public String showCart(Model

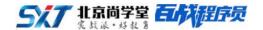
model,HttpServletRequest request){

model.addAttribute("cartList",

cartServiceImpl.showCart(request));

return "cart";

}
```



四.修改商品数量

1. 在 ego-cart 的 CartService 添加

```
/**
   * 根据 id 修改数量
   * @param id
   * @param num
   * @return
   */
  EgoResult update(long id,int num,HttpServletRequest
request);
@Override
  public EgoResult update(long id, int
num,HttpServletRequest request) {
     //redis 中的 key
     String token = CookieUtils.getCookieValue(request,
"TT TOKEN");
     String jsonUser =
HttpClientUtil.doPost(passportUrl+token);
     EgoResult er = JsonUtils.jsonToPojo(jsonUser,
EgoResult.class);
     String key =
```



```
cartKey+((LinkedHashMap)er.getData()).get("username")
;
     String json = jedisDaoImpl.get(key);
     List<TbItemChild> list =
JsonUtils.jsonToList(json, TbItemChild.class);
     for (TbItemChild child : list) {
        if((long)child.getId()==id){
           child.setNum(num);
        }
     }
     String ok = jedisDaoImpl.set(key,
JsonUtils.objectToJson(list));
     EgoResult egoResult = new EgoResult();
     if(ok.equals("OK")){
        egoResult.setStatus(200);
     }
     return egoResult;
  }
```

2. 在 ego-cart 中 CartController 添加

```
/**
* 修改商品数量
```



```
* @param id

* @param num

* @param request

* @return

*/

@RequestMapping("cart/update/num/{id}/{num}.action"
)

@ResponseBody

public EgoResult update(@PathVariable long
id,@PathVariable int num,HttpServletRequest request){
    return cartServiceImpl.update(id, num, request);
}
```

五. 删除购物车商品

1. 在 ego-cart 的 CartService

```
/**

* 删除购物车商品

* @param id

* @param req

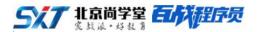
* @return

*/

EgoResult delete(long id, HttpServletRequest req);
```



```
@Override
  public EgoResult delete(long id, HttpServletRequest
req) {
     //redis 中的 key
     String token = CookieUtils.getCookieValue(reg,
"TT TOKEN");
     String jsonUser =
HttpClientUtil.doPost(passportUrl+token);
     EgoResult er = JsonUtils.jsonToPojo(jsonUser,
EgoResult.class);
     String key =
cartKey+((LinkedHashMap)er.getData()).get("username")
;
     String json = jedisDaoImpl.get(key);
     List<TbItemChild> list =
JsonUtils.jsonToList(json, TbItemChild.class);
     TbItemChild tbItemChild = null;
     for (TbItemChild child : list) {
        if((long)child.getId()==id){
           tbItemChild = child;
        }
```



```
}
list.remove(tbItemChild);
String ok = jedisDaoImpl.set(key,

JsonUtils.objectToJson(list));
EgoResult egoResult = new EgoResult();
if(ok.equals("OK")){
    egoResult.setStatus(200);
}
return egoResult;
}
```

2. 在 ego-cart 的 CartController

```
/**

* 删除购物车商品

* @param id

* @param req

* @return

*/

@RequestMapping("cart/delete/{id}.action")

@ResponseBody

public EgoResult delete(@PathVariable long

id,HttpServletRequest req){

return cartServiceImpl.delete(id, req);
```



}