

淘淘商城

第十一天

**讲师：传智.入云龙**

# 今日大纲

1. 实现淘淘商城的购物车功能
2. 实现订单系统。

# 购物车功能

## 功能说明

1. 商品加入购物车时，不是必须要求登录。
2. 计算购物车中商品的总价。当商品数量发生变化时需要重新计算。
3. 用户可以删除购物车中的商品。
4. 用户下单后，删除购物车的功能。

## 功能分析

1. 在用户不登陆的清空下也可以使用购物车，那么就需要把购物车信息放入cookie中。
2. 可以把商品信息，存放到pojo中，然后序列化成json存入cookie中。
3. 取商品信息可以从cookie中把json数据取出来，然后转换成java对象即可。
4. 此功能只需要操作cookie不需要数据库的支持，所以只需要在taotao-portal中实现即可。
5. 购物车分有四种动作
   1. 添加商品
   2. 修改商品数量
   3. 删除购物车中的商品
   4. 展示购物车商品列表

## 添加购物车商品

### service

**功能：**

1、接收controller传递过来的商品id，根据商品id查询商品信息。

2、从cookie中取出购物车信息，转换成商品pojo列表。

3、把商品信息添加到商品列表中。

**参数：**

1、商品id

2、Request

3、response

**返回值：**

TaoTaoResult

|  |
| --- |
| @Service  **public** **class** CartServiceImpl **implements** CartService {    //服务url  @Value("${REST\_BASE\_URL}")  **private** String REST\_BASE\_URL;  //商品服务url  @Value("${ITEMS\_ITEM\_URL}")  **private** String ITEMS\_ITEM\_URL;  //COOKIE中购物车商品对应的key  @Value("${CART\_ITEMS\_LIST\_KEY}")  **private** String CART\_ITEMS\_LIST\_KEY;  //购物车cookie生存期  @Value("${CART\_ITEMS\_EXPIRE\_TIME}")  **private** Integer CART\_ITEMS\_EXPIRE\_TIME;  /\*\*  \* 添加购物车商品  \* <p>Title: addItem</p>  \* <p>Description: </p>  \* **@param** itemId  \* **@param** request  \* **@param** response  \* **@return**  \* **@see** com.taotao.portal.service.CartService#addItem(java.lang.Long, javax.servlet.http.HttpServletRequest, javax.servlet.http.HttpServletResponse)  \*/  @Override  **public** TaotaoResult addItem(Long itemId, HttpServletRequest request, HttpServletResponse response) {  //根据商品id查询商品信息  Item item = getItemById(itemId);  **if** (item == **null**) **return** TaotaoResult.*build*(400, "未查询到该商品信息");  //取cookie中购物车商品列表  List<Item> cartItems = getItemListFromCookie(request);  //判断该商品是否存在于购物车中  **boolean** itemExists = **false**;  **for** (Item i : cartItems) {  **if** (i.getId().longValue() == itemId.longValue()) {  //购物车中有此商品  i.setNum(i.getNum() + 1);  itemExists = **true**;  **break**;  }    }  //如果商品不存在于购物车则向购物车商品列表中添加一个商品  **if** (! itemExists) {  //设置数量为1  item.setNum(1);  //把商品添加到购物车  cartItems.add(item);  }  //把购物车信息写入cookie中  CookieUtils.*setCookie*(request, response, CART\_ITEMS\_LIST\_KEY, JsonUtils.*objectToJson*(cartItems), CART\_ITEMS\_EXPIRE\_TIME, **true**);    **return** TaotaoResult.*ok*(cartItems);  }    **private** Item getItemById(Long itemId) {  //根据商品id查询商品信息  String resultStr = HttpClientUtil.*doGet*(REST\_BASE\_URL + ITEMS\_ITEM\_URL + itemId);  //转换成taotaoResult  TaotaoResult result = TaotaoResult.*formatToPojo*(resultStr, Item.**class**);  //取商品信息  Item item = **null**;  **if** (result.getStatus() == 200) {  item = (Item) result.getData();  }    **return** item;  }    **private** List<Item> getItemListFromCookie(HttpServletRequest request) {  //取cookie中购物车商品列表  String cartItemsStr = CookieUtils.*getCookieValue*(request, CART\_ITEMS\_LIST\_KEY, **true**);  //如果不为空那么就转换成java对象  List<Item> cartItems = **null**;  **if** (!StringUtils.*isBlank*(cartItemsStr)) {  cartItems = JsonUtils.*jsonToList*(cartItemsStr, Item.**class**);  } **else** {  cartItems = **new** ArrayList<>();  }  **return** cartItems;  }  } |

### 商品pojo

商品pojo需要使用taotao-portal中的Item。此pojo在反序列化时会抛异常。需要做如下修改：



### Controller

|  |
| --- |
| @Controller  @RequestMapping("/cart")  **public** **class** CartController {  @Autowired  **private** CartService cartService;    @RequestMapping("/add/{itemId}")  **public** String addItem(@PathVariable Long itemId,  HttpServletRequest request, HttpServletResponse response, Model model) {  //添加商品信息  TaotaoResult result = cartService.addItem(itemId, request, response);  //错误信息  **if** (result.getStatus() != 200) {  model.addAttribute("message", result.getMsg());  **return** "error/exception";  }  //把购物车中的商品传递给页面  model.addAttribute("cartList", result.getData());  **return** "cart";  }    } |

## 展示购物车商品

用户点击“我的购物车”展示购物车信息

### Service

|  |
| --- |
| /\*\*  \* 取购物车列表  \* <p>  \* Title: getCartItemsList  \* </p>  \* <p>  \* Description:  \* </p>  \*  \* **@return**  \* **@see** com.taotao.portal.service.CartService#getCartItemsList()  \*/  @Override  **public** List<Item> getCartItemsList(HttpServletRequest request) {  // 从cookie中取商品列表  List<Item> itemsList = getItemListFromCookie(request);  **return** itemsList;  } |

### Controller

|  |
| --- |
| @RequestMapping("/cart")  **public** String showCart(HttpServletRequest request, Model mode) {  //取购物车信息  List<Item> list = cartService.getCartItemsList(request);  mode.addAttribute("cartList", list);  **return** "cart";  } |

## 修改商品数量

当点击购物车商品的“+”、“-”号时增加或减少商品数量。减少商品数量时，如果数量为“1”则不继续减少。

### Service

|  |
| --- |
| /\*\*  \* 修改指定商品的数量  \* <p>Title: changeItemNum</p>  \* <p>Description: </p>  \* **@param** itemId  \* **@param** num  \* **@param** request  \* **@param** response  \* **@return**  \* **@see** com.taotao.portal.service.CartService#changeItemNum(long, int, javax.servlet.http.HttpServletRequest, javax.servlet.http.HttpServletResponse)  \*/  @Override  **public** TaotaoResult changeItemNum(**long** itemId, **int** num, HttpServletRequest request, HttpServletResponse response) {  //从cookie中取商品列表  List<Item> list = getItemListFromCookie(request);  //从商品列表中找到要修改数量的商品  **for** (Item item : list) {  **if** (item.getId() == itemId) {  //找到商品，修改数量  item.setNum(num);  **break**;  }  }  //把商品信息写入cookie  CookieUtils.*setCookie*(request, response, CART\_ITEMS\_LIST\_KEY, JsonUtils.*objectToJson*(list), CART\_ITEMS\_EXPIRE\_TIME, **true**);    **return** TaotaoResult.*ok*();  } |

### Controller

|  |
| --- |
| @RequestMapping("/update/num/{itemId}/{num}")  @ResponseBody  **public** TaotaoResult updateNumById(@PathVariable Long itemId, @PathVariable Integer num,  HttpServletRequest request, HttpServletResponse response) {  TaotaoResult result = cartService.changeItemNum(itemId, num, request, response);  **return** result;  } |

## 删除购物车商品

### Service

|  |
| --- |
| /\*\*  \* 删除购物车中的商品  \* <p>  \* Title: deleteItem  \* </p>  \* <p>  \* Description:  \* </p>  \*  \* **@param** itemId  \* **@param** request  \* **@param** response  \* **@return**  \* **@see** com.taotao.portal.service.CartService#deleteItem(java.lang.Long,  \* javax.servlet.http.HttpServletRequest,  \* javax.servlet.http.HttpServletResponse)  \*/  @Override  **public** List<Item> deleteItem(Long itemId, HttpServletRequest request, HttpServletResponse response) {  List<Item> itemsList = getCartItemsList(request);  // 找到购物车中的商品，并删除之  **for** (Item item : itemsList) {  **if** (item.getId().longValue() == itemId.longValue()) {  itemsList.remove(item);  **break**;  }  }  // 更新cookie中的购物车数据  CookieUtils.*setCookie*(request, response, CART\_ITEMS\_LIST\_KEY, JsonUtils.*objectToJson*(itemsList), CART\_ITEMS\_EXPIRE\_TIME, **true**);  **return** itemsList;  } |

### Controller

|  |
| --- |
| @RequestMapping("/delete/{itemId}")  **public** String deleteItemById(@PathVariable Long itemId,  HttpServletRequest request, HttpServletResponse response,  Model model) {  List<Item> list = cartService.deleteItem(itemId, request, response);  model.addAttribute("cartList", list);  **return** "cart";  } |

# 订单系统实现

## 系统架构

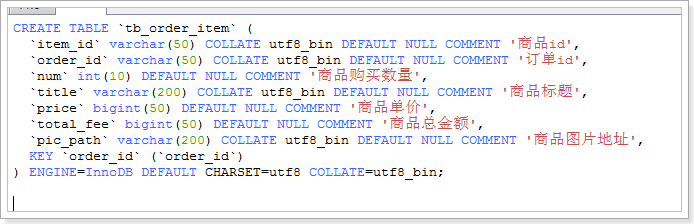


## 订单表结构

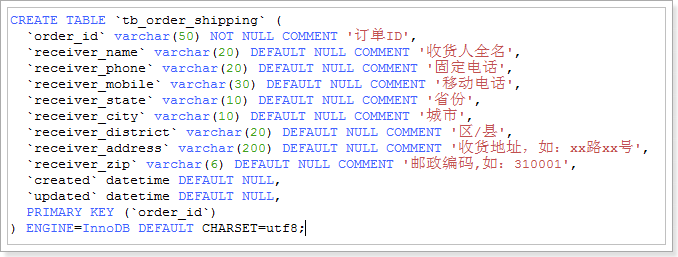
订单表：



订单商品表：



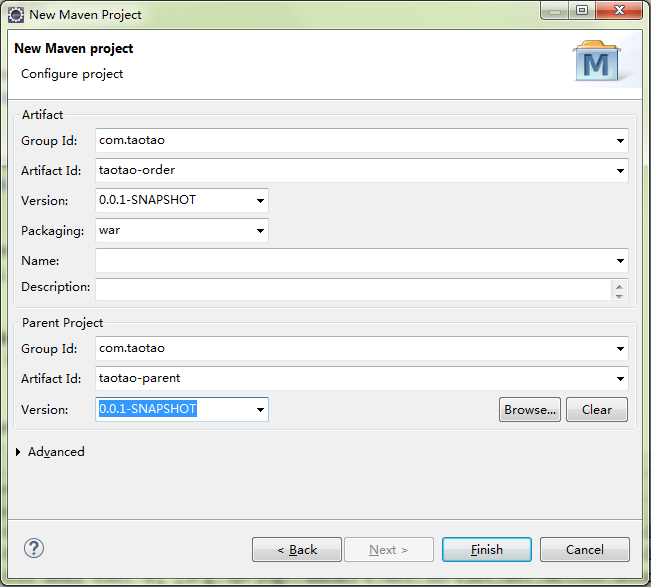
物流表：



## 接口

参见：淘淘商城-订单系统接口.docx

## 创建订单系统taotao-order



系统配置参考taotao-sso系统。

## 实现创建订单接口

### service

|  |
| --- |
| @Service  **public** **class** OrderServiceImpl **implements** OrderService {  @Autowired  **private** TbOrderMapper orderMapper;  @Autowired  **private** TbOrderItemMapper orderItemMapper;  @Autowired  **private** TbOrderShippingMapper orderShippingMapper;  //redis中的订单key  @Value("${ORDER\_ID\_KEY}")  **private** String ORDER\_ID\_KEY;  @Value("${ORDER\_BEGIN\_ID}")  **private** Long ORDER\_BEGIN\_ID;    @Autowired  **private** JedisCluster jedisCluster;    @Override  **public** TaotaoResult createOrder(TbOrder order, List<TbOrderItem> itemList, TbOrderShipping orderShipping) {    //订单表  //生成订单号  String orderIdStr = jedisCluster.get(ORDER\_ID\_KEY);  Long orderId = **null**;  **if** (StringUtils.*isBlank*(orderIdStr)) {  //如果redis中没有订单号使用初始订单号初始化  jedisCluster.set(ORDER\_ID\_KEY, ORDER\_BEGIN\_ID.toString());  orderId = ORDER\_BEGIN\_ID;  } **else** {  //生成订单号  orderId = jedisCluster.incr(ORDER\_ID\_KEY);  }  //设置订单号  order.setOrderId(orderId.toString());  Date date = **new** Date();  //订单创建时间  order.setCreateTime(date);  //订单更新时间  order.setUpdateTime(date);  //插入订单表  orderMapper.insert(order);  //插入订单商品表  **for** (TbOrderItem tbOrderItem : itemList) {  //取订单商品id  Long orderItemId = jedisCluster.incr("ORDER\_ITEM\_ID");  tbOrderItem.setId(orderItemId.toString());  tbOrderItem.setOrderId(orderId.toString());  //添加到订单商品表  orderItemMapper.insert(tbOrderItem);  }  //插入物流表  orderShipping.setOrderId(orderId.toString());  orderShipping.setCreated(date);  orderShipping.setUpdated(date);  orderShippingMapper.insert(orderShipping);    **return** TaotaoResult.*ok*(orderId.toString());  }  } |

### Controller

#### Pojo

|  |
| --- |
| **public** **class** Order **extends** TbOrder{  **private** List<TbOrderItem> orderItems;  **private** TbOrderShipping orderShipping;  } |

#### controller

|  |
| --- |
| @Controller  @RequestMapping("/order")  **public** **class** OrderController {  @Autowired  **private** OrderService orderService;    @RequestMapping(value = "/create", method=RequestMethod.***POST***)  @ResponseBody  **public** TaotaoResult createOrder(@RequestBody Order order) {  TaotaoResult result = **null**;  **try** {  result = orderService.createOrder(order, order.getOrderItems(), order.getOrderShipping());  } **catch** (Exception e) {  e.printStackTrace();  **return** TaotaoResult.*build*(500, ExceptionUtil.*getStackTrace*(e));  }  **return** result;  }  } |

## 其他接口

课后作业

# 前端系统实现下单功能

## 下单流程



## 订单结算页面

### Controller

|  |
| --- |
| @Controller  @RequestMapping("/order")  **public** **class** OrderController {  @Autowired  **private** OrderService orderService;  @Autowired  **private** CartService cartService;      /\*\*  \* 显示订单确认画面  \* <p>Title: showOrderCart</p>  \* <p>Description: </p>  \* **@param** request  \* **@param** model  \* **@return**  \*/  @RequestMapping("/order-cart")  **public** String showOrderCart(HttpServletRequest request, TbUser user, Model model) {  //根据用户信息，取出用户的收货地址列表  //本项目中使用静态数据模拟。。。。    //从cookie中把商品列表取出来  List<Item> itemsList = cartService.getCartItemsList(request);  model.addAttribute("cartList", itemsList);  **return** "order-cart";  }  } |

## 提交订单处理

### 创建订单Service

|  |
| --- |
| @Srvice  **public** **class** OrderServiceImpl **implements** OrderService {  @Value("${ORDER\_BASE\_URL}")  **private** String ORDER\_BASE\_URL;    @Value("${ORDER\_CREATE\_URL}")  **private** String ORDER\_CREATE\_URL;      @Override  **public** TaotaoResult createService(Order order) {    //把pojo转换成json数据  String json = JsonUtils.*objectToJson*(order);  //调用订单系统服务提交订单  String resultStr = HttpClientUtil.*doPostJson*(ORDER\_BASE\_URL + ORDER\_CREATE\_URL, json);  //转换成java对象  TaotaoResult taotaoResult = TaotaoResult.*format*(resultStr);    **return** taotaoResult;  }  } |

### Controller

|  |
| --- |
| @RequestMapping(value="/create", method=RequestMethod.***POST***)  **public** String createOrder(HttpServletRequest request, Order order, Model model) {  //从request中取用户信息  TbUser user = (TbUser)request.getAttribute("user");  //把用户信息补充到order对象中  order.setUserId(user.getId());  order.setBuyerNick(user.getUsername());    //提交订单  TaotaoResult result = **null**;  **try** {  result = orderService.createService(order);  //订单创建成功  **if**(result.getStatus() == 200) {  model.addAttribute("orderId", result.getData());  model.addAttribute("payment", order.getPayment());  //两天后送达  DateTime dateTime = **new** DateTime();  dateTime = dateTime.plusDays(2);  model.addAttribute("date", dateTime.toString("yyyy-MM-dd"));  **return** "success";  }  } **catch** (Exception e) {  e.printStackTrace();  result = TaotaoResult.*build*(500, ExceptionUtil.*getStackTrace*(e));  }  //订单创建失败  model.addAttribute("message", result.getMsg());  **return** "error/exception";    } |