**品优购电商系统开发**

**第16章**

**购物车解决方案**

传智播客.黑马程序员

# 课程目标

目标1：明白购物车整体思路

目标2：运用Cookie存储购物车

目标3：编写购物车前端代码

目标4：掌握Redis存储购物车实现

# 购物车需求分析与解决方案

## 1.1需求分析

用户在商品详细页点击加入购物车，提交商品SKU编号和购买数量，添加到购物车。

思考的问题：

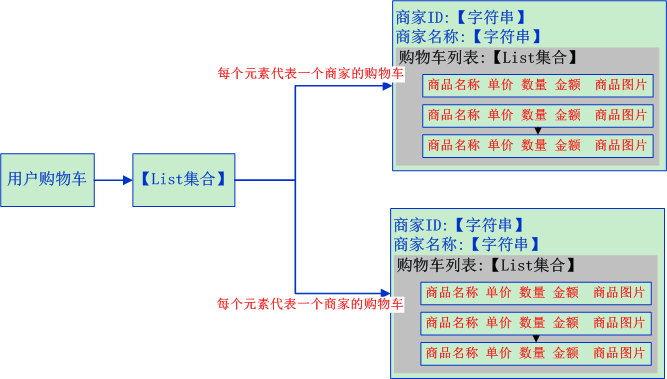
品优购是一个B2B2C的电商平台，有商家入驻，平台需要与商家月底结算，那么用户在添加商品至购物车时，就应该按不同商家添加。这样一个购物车包含多个商家，一个商家又包含多个商品。

购物车展示页面如下：



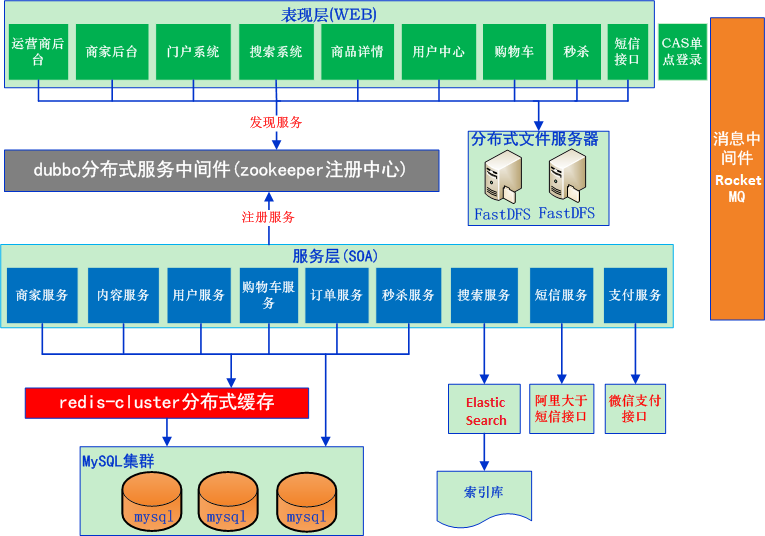
## 1.2实现思路

* 定义购物车数据的存储结构



* 当用户在未登录的情况下，将购物车数据存入Cookie。(用户浏览器)
* 当用户在登录的情况下，将购物车数据存入Redis。(适合长时间存储，读写速度快)
* 如果用户登录时，需要将Cookie中的购物车数据合并到Redis中存储，并删除Cookie中的购物车数据。

## 1.3搭建购物车工程



### 1.3.1 pinyougou-cart-service(war)

与pinyougou-sms-service服务层的构建方式一样，拷贝pinyougou-sms-service模块中的web.xml、applicationContext-service.xml、log4j.properties。

pinyougou-cart-service/pom.xml：

|  |
| --- |
| <**properties**>  *<!-- 指定端口号 -->* <**port**>9006</**port**> </**properties**> |

pinyougou-cart-service/src/main/resources/applicationContext-service.xml：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"  xmlns:context="http://www.springframework.org/schema/context"  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd  http://code.alibabatech.com/schema/dubbo  http://code.alibabatech.com/schema/dubbo/dubbo.xsd  http://www.springframework.org/schema/context  http://www.springframework.org/schema/context/spring-context.xsd"**>  *<!-- 配置加载属性文件 -->* <**context:property-placeholder  location="classpath\*:props/\*.properties"**/>  *<!-- 导入数据访问配置文件 -->* <**import resource="classpath:applicationContext-mapper.xml"**/>  *<!-- 导入redis配置文件 -->* <**import resource="classpath:applicationContext-redis.xml"**/>  *<!-- ############## 配置dubbo服务提供者 ############## -->  <!-- 配置当前应用的名称 -->* <**dubbo:application name="pinyougou-cart-service"**/>  *<!-- 配置zookeeper作为注册中心，注册服务地址 -->* <**dubbo:registry protocol="zookeeper" address="192.168.12.131:2181"**/>  *<!-- 用dubbo协议在20885端口暴露服务 -->* <**dubbo:protocol name="dubbo" port="20885"**/>  *<!-- 配置采用包扫描来暴露服务 -->* <**dubbo:annotation package="com.pinyougou.cart.service.impl"**/> </**beans**> |

### 1.3.2 pinyougou-cart-web(war)

与pinyougou-sms-web构建方式一样，拷贝pinyougou-sms-web模块中的web.xml、springmvc.xml、log4j.properties

pinyougou-cart-web/pom.xml：

|  |
| --- |
| <**properties**>  *<!-- tomcat端口号 -->* <**port**>**9109**</**port**> </**properties**> |

pinyougou-cart-web/src/main/webapp/WEB-INF/web.xml：

|  |
| --- |
| *<!-- 欢迎文件列表 -->*  <**welcome-file-list**>  <**welcome-file**>cart.html</**welcome-file**> </**welcome-file-list**> |

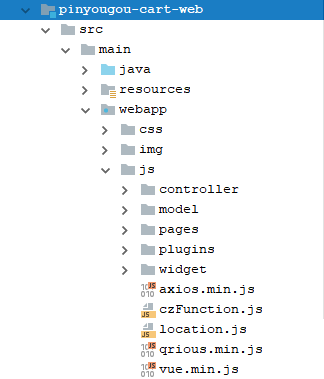
pinyougou-cart-web/src/main/resources/springmvc.xml：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns:mvc="http://www.springframework.org/schema/mvc"  xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"  xmlns:context="http://www.springframework.org/schema/context"  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd  http://www.springframework.org/schema/mvc  http://www.springframework.org/schema/mvc/spring-mvc.xsd  http://code.alibabatech.com/schema/dubbo  http://code.alibabatech.com/schema/dubbo/dubbo.xsd  http://www.springframework.org/schema/context  http://www.springframework.org/schema/context/spring-context.xsd"**>    *<!-- 配置开启组件扫描 -->* **<context:component-scan base-package="com.pinyougou.cart.controller"/>**  *<!-- 开启MVC注解驱动 -->* <**mvc:annotation-driven**>  *<!-- 配置消息转换 -->* <**mvc:message-converters**>  *<!-- 配置用fastjson作为json处理框架 -->* <**bean class="com.alibaba.fastjson.support.spring.FastJsonHttpMessageConverter"**>  *<!-- 配置支持的类型 -->* <**property name="supportedMediaTypes"  value="application/json;charset=UTF-8"**/>  </**bean**>  </**mvc:message-converters**>  </**mvc:annotation-driven**>   *<!-- 配置静态资源用WEB容器默认的servlet来处理 -->* <**mvc:default-servlet-handler**/>   *<!-- ############## 配置dubbo服务消费者 ############## -->  <!-- 配置当前应用的名称 -->* <**dubbo:application name="pinyougou-cart-web"**/>  *<!-- 配置zookeeper作为注册中心，注册服务地址 -->* <**dubbo:registry protocol="zookeeper" address="192.168.12.131:2181"**/>  *<!-- 配置采用包扫描来引用服务，产生服务接口的代理对象 -->* <**dubbo:annotation package="com.pinyougou.cart.controller"**/> </**beans**> |

## 1.4准备工作

### 1.4.1静态资源

拷贝【资料\购物车系统静态资源】到pinyougou-cart-web/src/main/webapp/目录：



### 1.4.2 Cookie工具类

拷贝【资料】中的CookieUtils.java工具类到pinyougou-common工程的com.pinyougou.common.util包下，需要在pinyougou-common/pom.xml引入servlet-api依赖：

|  |
| --- |
| *<!-- servlet-api -->* <**dependency**>  <**groupId**>org.apache.tomcat.embed</**groupId**>  <**artifactId**>tomcat-embed-core</**artifactId**>  <**scope**>provided</**scope**> </**dependency**> |

### 1.4.3购物车实体

pinyougou-pojo/src/main/java/com/pinyougou/cart/包下，创建购物车实体类Cart.java：

|  |
| --- |
| */\*\* 购物车实体 \*/* **public class** Cart **implements** Serializable{  */\*\* 商家ID \*/* **private** String **sellerId**;  */\*\* 商家名称 \*/* **private** String **sellerName**;  */\*\* 购物车明细集合 \*/* **private** List<OrderItem> **orderItems**;  */\*\* setter and getter method \*/* |

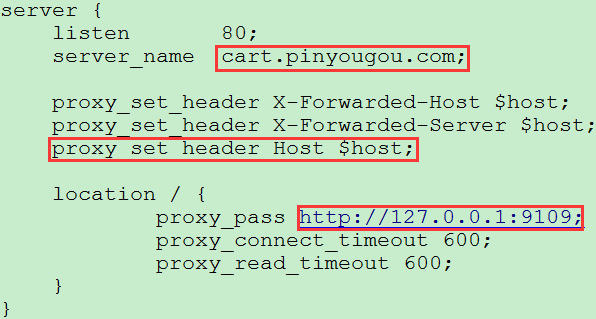
**说明：该实体类是对每个商家的购物车进行的封装。**

### 1.4.4配置域名访问

修改hosts：127.0.0.1 cart.pinyougou.com

修改nginx.conf：

比前多增加一个配置设置携带域名：proxy\_set\_header Host $host;



# 购物车系统集成CAS

## 2.1配置集成CAS

pinyougou-cart-web/pom.xml，配置依赖：

|  |
| --- |
| <**dependencies**>  *<!-- cas-client-core -->* <**dependency**>  <**groupId**>org.jasig.cas.client</**groupId**>  <**artifactId**>cas-client-core</**artifactId**>  </**dependency**> </**dependencies**> |

pinyougou-cart-web/src/main/webapp/WEB-INF/web.xml，增加CAS过滤器：

|  |
| --- |
| *<!--############## CAS单点登录配置 开始 ################-->  <!-- 配置单点退出过滤器 (可选配置) -->* <**filter**>  <**filter-name**>ssoutFilter</**filter-name**> <**filter-class**>org.jasig.cas.client.session.SingleSignOutFilter</**filter-class**>  *<!-- 配置CAS服务URL前缀 -->* <**init-param**>  <**param-name**>casServerUrlPrefix</**param-name**>  <**param-value**>http://sso.pinyougou.com</**param-value**>  </**init-param**>  </**filter**>  <**filter-mapping**>  <**filter-name**>ssoutFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**>  </**filter-mapping**>   *<!-- 配置身份认证过滤器(不需要跳转到CAS登录系统) -->* <**filter**>  <**filter-name**>anonAuthenticationFilter</**filter-name**> <**filter-class**>org.jasig.cas.client.authentication.AuthenticationFilter</**filter-class**>  <**init-param**>  *<!-- 配置CAS服务端登录地址 -->* <**param-name**>casServerLoginUrl</**param-name**>  <**param-value**>http://sso.pinyougou.com</**param-value**>  </**init-param**>  <**init-param**>  *<!-- 配置服务名称 -->* <**param-name**>serverName</**param-name**>  <**param-value**>http://cart.pinyougou.com</**param-value**>  </**init-param**>  *<!-- 配置gateway=true”时，启用异步方式认证 -->* <**init-param**>  <**param-name**>gateway</**param-name**>  <**param-value**>true</**param-value**>  </**init-param**>  </**filter**>  <**filter-mapping**>  <**filter-name**>anonAuthenticationFilter</**filter-name**>  <**url-pattern**>\*.html</**url-pattern**>  </**filter-mapping**>   *<!-- 配置身份认证过滤器(需要跳转到CAS登录系统) -->* <**filter**>  <**filter-name**>authenticationFilter</**filter-name**> <**filter-class**>org.jasig.cas.client.authentication.AuthenticationFilter</**filter-class**>  <**init-param**>  *<!-- 配置CAS服务端登录地址 -->* <**param-name**>casServerLoginUrl</**param-name**>  <**param-value**>http://sso.pinyougou.com</**param-value**>  </**init-param**>  <**init-param**>  *<!-- 配置服务名称-->* <**param-name**>serverName</**param-name**>  <**param-value**>http://cart.pinyougou.com</**param-value**>  </**init-param**>  </**filter**>  <**filter-mapping**>  <**filter-name**>authenticationFilter</**filter-name**>  <**url-pattern**>/order/\*</**url-pattern**>  </**filter-mapping**>   *<!-- 该过滤器负责对ticket的校验工作(必须启用) -->* <**filter**>  <**filter-name**>ticketValidationFilter</**filter-name**> <**filter-class**>org.jasig.cas.client.validation.Cas20ProxyReceivingTicketValidationFilter</**filter-class**>  *<!-- 配置CAS服务URL前缀 -->* <**init-param**>  <**param-name**>casServerUrlPrefix</**param-name**>  <**param-value**>http://sso.pinyougou.com</**param-value**>  </**init-param**>  *<!-- 配置服务名称 -->* <**init-param**>  <**param-name**>serverName</**param-name**>  <**param-value**>http://cart.pinyougou.com</**param-value**>  </**init-param**>  </**filter**>  <**filter-mapping**>  <**filter-name**>ticketValidationFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**>  </**filter-mapping**>   *<!--  配置HttpServletRequest请求包裹过滤器 (可选配置)  说明：通过HttpServletRequest的getRemoteUser()方法获得SSO登录用户名  -->* <**filter**>  <**filter-name**>httpServletRequestWrapperFilter</**filter-name**> <**filter-class**>org.jasig.cas.client.util.HttpServletRequestWrapperFilter</**filter-class**>  </**filter**>  <**filter-mapping**>  <**filter-name**>httpServletRequestWrapperFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**>  </**filter-mapping**>  *<!--############## CAS单点登录配置 结束 ################-->* |

## 2.2登录用户名与退出

pinyougou-cart-web/src/main/java/com/pinyougou/cart/controller/LoginController.java：

|  |
| --- |
| @RestController **public class** LoginController {  */\*\* 显示用户名 \*/*  @GetMapping(**"/user/showName"**)  **public** Map<String,String> showName(HttpServletRequest request){  */\*\* 获取登录用户名 \*/* String name = request.getRemoteUser();  Map<String, String> map = **new** HashMap<>();  map.put(**"loginName"**, name);  **return** map;  } } |

pinyougou-cart-web\src\main\webapp\js\controller\cartController.js：

|  |
| --- |
| *// 窗口加载完* ***window***.onload = **function** () {  **var** vue = **new Vue**({  **el** : **'#app'**, *// 元素绑定* **data** : { *// 数据模型* **loginName** : **''**, *// 登录用户名* **redirectUrl** : **''** *// 重定向URL* },  **methods** : { *// 操作方法  // 加载用户* loadUsername : **function** () {  *// 定义重定向URL* **this**.**redirectUrl** = ***window***.*encodeURIComponent*(***location***.href);  *// 获取登录用户名* **axios**.get(**"/user/showName"**).then(**function**(response){  vue.**loginName** = response.**data**.**loginName**;  });  }  },  created : **function** () { *// 创建生命周期* **this**.loadUsername();  }  }); }; |

修改pinyougou-cart-web/src/main/webapp/cart.html页面：

|  |
| --- |
| <**script src="/js/vue.min.js"**></**script**> <**script src="/js/axios.min.js"**></**script**> <**script src="/js/controller/cartController.js"**></**script**>  <**ul class="fl"**>  <**li class="f-item" v-if="loginName"**>  {{**loginName**}}，品优购欢迎您！**&nbsp;** <**a :href="'http://sso.pinyougou.com/logout?service='+redirectUrl"**>退出</**a**>  </**li**>  <**li class="f-item" v-if="!loginName"**>请  <**a :href="'http://sso.pinyougou.com/?service='+redirectUrl"**>登录</**a**>  <**a href="http://user.pinyougou.com/register.html"**>免费注册</**a**>  </**li**> </**ul**> |

# 3.Cookie存储购物车【操作Cookie】

## 3.1需求分析

使用Cookie存储购物车数据，服务层负责业务逻辑，控制器层负责读写Cookie。

说明：Cookie的value存储List<Cart>的json格式字符串数据(中文需要unicode编码，因为Cookie的值不能存在中文字符)。

## 3.2控制器层

实现思路分析：

* 从Cookie中获取购物车
* 往购物车中添加商品
* 将购物车存入Cookie

pinyougou-cart-web/src/main/java/com/pinyougou/cart/controller/包下，创建CartController.java控制器：

|  |
| --- |
| */\*\* 购物车控制器 \*/* @RestController @RequestMapping(**"/cart"**) **public class** CartController {  @Reference(timeout = 30000)  **private** CartService **cartService**;  @Autowired  **private** HttpServletRequest **request**;  @Autowired  **private** HttpServletResponse **response**;  */\*\* 添加SKU商品到购物车 \*/* @GetMapping(**"/addCart"**)  **public boolean** addCart(Long itemId, Integer num){  **try**{  *// 获取购物车集合* List<Cart> carts = findCart();  *// 调用服务层添加SKU商品到购物车* carts = **cartService**.addItemToCart(carts, itemId, num);  *// 将购物车重新存入Cookie中* CookieUtils.*setCookie*(**request**, **response**,  CookieUtils.CookieName.***PINYOUGOU\_CART***,  JSON.*toJSONString*(carts),  3600 \* 24, **true**);  **return true**;  }**catch** (Exception ex){  ex.printStackTrace();  }  **return false**;  }  */\*\* 获取购物车集合 \*/* @GetMapping(**"/findCart"**)  **public** List<Cart> findCart() {  *// 从Cookie中获取购物车集合json字符串* String cartStr = CookieUtils.*getCookieValue*(**request**,  CookieUtils.CookieName.***PINYOUGOU\_CART***, **true**);  *// 判断是否为空* **if** (StringUtils.*isBlank*(cartStr)){  cartStr = **"[]"**;  }  List<Cart> carts = JSON.*parseArray*(cartStr, Cart.**class**);  **return** carts;  } } |

## 3.3服务接口层

pinyougou-interface\src\main\java\com\pinyougou\service\包下，创建服务接口CartService.java：

|  |
| --- |
| */\*\* 购物车服务接口 \*/* **public interface** CartService {  */\*\*  \* 添加SKU商品到购物车  \** ***@param carts*** *购物车(一个Cart对应一个商家)  \** ***@param itemId*** *SKU商品id  \** ***@param num*** *购买数据  \** ***@return*** *修改后的购物车  \*/* List<Cart> addItemToCart(List<Cart> carts,  Long itemId, Integer num); } |

## 3.4服务实现层

实现思路：

|  |
| --- |
| 1.根据SKU商品ID查询SKU商品对象  2.获取商家ID  3.根据商家ID判断购物车集合中是否存在该商家的购物车  4.如果购物车集合中不存在该商家的购物车  4.1 创建新的购物车对象  4.2 将新的购物车对象添加到购物车集合  5.如果购物车集合中存在该商家的购物车  5.1 判断购物车订单明细集合中是否存在该商品  5.2 如果没有，新增购物车订单明细  5.3 如果有，在原购物车订单明细上添加数量，更改金额 |

pinyougou-cart-service/src/main/java/com.pinyougou.cart.service.impl包下，创建服务接口实现类CartServiceImpl.java：

|  |
| --- |
| */\*\* 购物车服务接口实现类 \*/* @Service(interfaceName = **"com.pinyougou.cart.service.CartService"**) @Transactional **public class** CartServiceImpl **implements** CartService {  @Autowired  **private** ItemMapper **itemMapper**;  */\*\*  \* 添加SKU商品到购物车  \** ***@param carts*** *购物车(一个Cart对应一个商家)  \** ***@param itemId*** *SKU商品id  \** ***@param num*** *购买数据  \** ***@return*** *修改后的购物车  \*/* **public** List<Cart> addItemToCart(List<Cart> carts,  Long itemId, Integer num){  **try**{  *// 根据SKU商品ID查询SKU商品对象* Item item = **itemMapper**.selectByPrimaryKey(itemId);  *// 获取商家id* String sellerId = item.getSellerId();  *// 根据商家id判断购物车集合中是否存在该商家的购物车* Cart cart = searchCartBySellerId(carts, sellerId);  **if** (cart == **null**){ *// 购物车集合中不存在该商家购物车  // 创建新的购物车对象* cart = **new** Cart();  cart.setSellerId(sellerId);  cart.setSellerName(item.getSeller());  *// 创建订单明细(购物中一个商品)* OrderItem orderItem = createOrderItem(item, num);  List<OrderItem> orderItems = **new** ArrayList<>(0);  orderItems.add(orderItem);  *// 为购物车设置订单明细集合* cart.setOrderItems(orderItems);   *// 将新的购物车对象添加到购物车集合* carts.add(cart);  }**else**{ *// 购物车集合中存在该商家购物车  // 判断购物车订单明细集合中是否存在该商品* OrderItem orderItem = searchOrderItemByItemId(  cart.getOrderItems(), itemId);  **if** (orderItem == **null**){  *// 如果没有，新增购物车订单明细* orderItem = createOrderItem(item, num);  cart.getOrderItems().add(orderItem);  }**else**{  *// 如果有，在原购物车订单明细上添加数量，更改金额* orderItem.setNum(orderItem.getNum() + num);  orderItem.setTotalFee(**new** BigDecimal(  orderItem.getPrice().doubleValue()  \* orderItem.getNum()));  *// 如果订单明细的购买数小于等于0，则删除* **if** (orderItem.getNum() <= 0){  *// 删除购物车中的订单明细(商品)* cart.getOrderItems().remove(orderItem);  }  *// 如果cart的orderItems订单明细为0，则删除cart* **if** (cart.getOrderItems().size() == 0){  carts.remove(cart);  }  }  }  **return** carts;  }**catch** (Exception ex){  **throw new** RuntimeException(ex);  }  }  */\*\* 从购物车集合中获取该商家的购物车 \*/* **private** Cart searchCartBySellerId(  List<Cart> carts, String sellerId) {  **for** (Cart cart : carts){  **if** (cart.getSellerId().equals(sellerId)){  **return** cart;  }  }  **return null**;  }  */\*\* 创建订单明细 \*/* **private** OrderItem createOrderItem(Item item, Integer num) {  *// 创建订单明细* OrderItem orderItem = **new** OrderItem();  orderItem.setSellerId(item.getSellerId());  orderItem.setGoodsId(item.getGoodsId());  orderItem.setItemId(item.getId());  orderItem.setNum(num);  orderItem.setPicPath(item.getImage());  orderItem.setPrice(item.getPrice());  orderItem.setTitle(item.getTitle());  *// 小计* orderItem.setTotalFee(**new** BigDecimal(  item.getPrice().doubleValue() \* num));  **return** orderItem;  }  */\*\* 从订单明细集合中获取指定订单明细 \*/* **private** OrderItem searchOrderItemByItemId(  List<OrderItem> orderItems, Long itemId) {  **for** (OrderItem orderItem : orderItems){  **if** (orderItem.getItemId().equals(itemId)){  **return** orderItem;  }  }  **return null**;  } } |

**添加购物车：[http://cart.pinyougou.com/cart/addCart?itemId=1369458&num=10](http://cart.pinyougou.com/cart/addCart?itemId=1369280&num=100)**

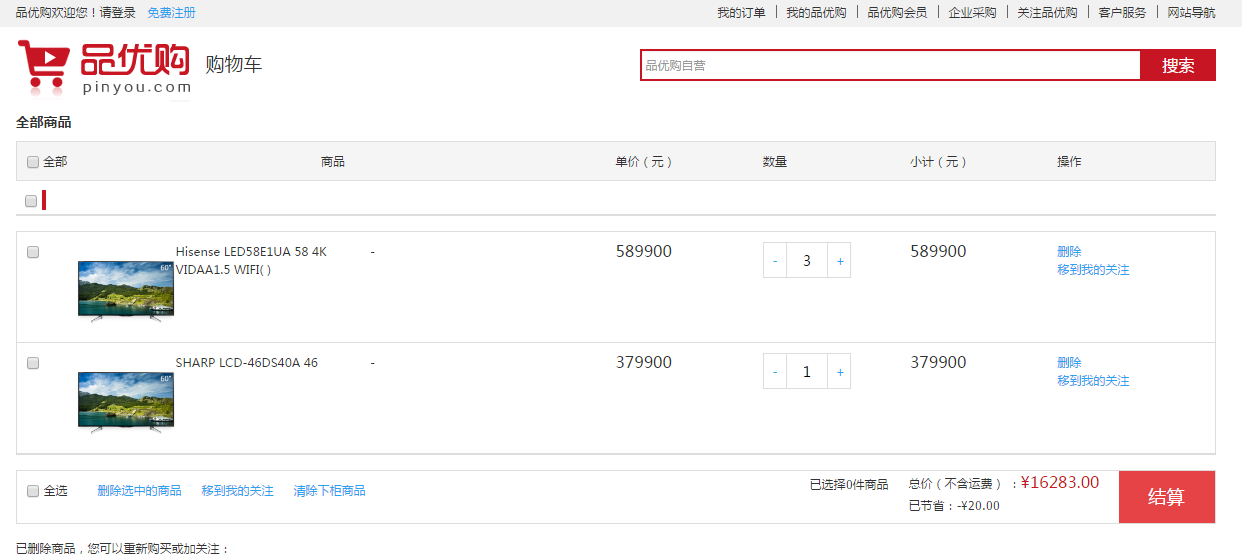
**查看购物车：<http://cart.pinyougou.com/cart/findCart>**

# 4.购物车前端代码

## 4.1需求分析

页面功能：购物车列表显示、数量的增减、删除商品、合计购买总数、合计购买总金额。

实现购物车页面的展示与相关操作：

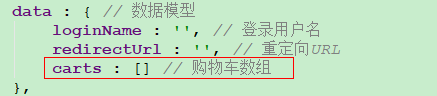


## 4.2购物车列表显示

### 4.2.1前端控制器层

pinyougou-cart-web/src/main/webapp/js/controller/cartController.js文件:

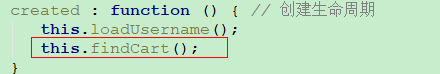
* 在data中定义变量



* 在methods中定义方法

|  |
| --- |
| *// 查询购物车* findCart : **function** () {  **axios**.get(**"/cart/findCart"**).then(**function**(response){  vue.**carts** = response.**data**;  }); } |

* 初始化调用



### 4.2.2页面

pinyougou-cart-web/src/main/webapp/cart.html循环显示购物车列表(77行-169行)：

|  |
| --- |
| <**div class="cart-item-list" v-for="cart in carts"**>  <**div class="cart-shop"**>  <**input type="checkbox"**/>  <**span class="shopname self"**>{{cart.sellerName}}</**span**>  </**div**>  <**div class="cart-body"**>  <**div class="cart-list"  v-for="orderItem in cart.orderItems"**>  <**ul class="goods-list yui3-g"**>  <**li class="yui3-u-1-24"**>  <**input type="checkbox"**/>  </**li**>  <**li class="yui3-u-11-24"**>  <**div class="good-item"**>  <**div class="item-img"**>  <**img :src="orderItem.picPath"**/>  </**div**>  <**div class="item-msg"**>   **{{orderItem.title}}**  </**div**>  </**div**>  </**li**>  <**li class="yui3-u-1-8"**>  <**span class="price"**>**{{orderItem.price.toFixed(2)}}**</**span**>  </**li**>  <**li class="yui3-u-1-8"**>  <**a href="javascript:void(0)" class="increment mins"**>-</**a**>  <**input type="text"  v-model="orderItem.num"  class="itxt"**/>  <**a href="javascript:void(0)" class="increment plus"**>+</**a**>  </**li**>  <**li class="yui3-u-1-8"**>  <**span class="sum"**>**{{orderItem.totalFee.toFixed(2)}}**</**span**>  </**li**>  <**li class="yui3-u-1-8"**>  <**a href="#none"**>删除</**a**><**br** />  <**a href="#none"**>移到我的关注</**a**>  </**li**>  </**ul**>  </**div**>  </**div**> </**div**> |

## 4.3购物车数量增减与删除

### 4.3.1前端控制层

pinyougou-cart-web/src/main/webapp/js/controller/cartController.js，在methods中定义方法：

|  |
| --- |
| *// 加入购物车* addCart : **function** (itemId, num) {  **axios**.get(**"/cart/addCart?itemId="** + itemId + **"&num="** + num)  .then(**function**(response){  **if** (response.**data**){  *// 重新加载购物车* vue.findCart();  }**else**{  *alert*(**"操作失败！"**);  }  }); } |

### 4.3.2页面

修改pinyougou-cart-web/src/main/webapp/cart.html实现数量增减(102行)：

|  |
| --- |
| <**li class="yui3-u-1-8"**>  <**a href="javascript:void(0)"  class="increment mins"  @click="addCart(orderItem.itemId,-1)"**>-</**a**>  <**input type="text"  v-model="orderItem.num"  class="itxt"**/>  <**a href="javascript:void(0)"  class="increment plus"  @click="addCart(orderItem.itemId,1)"**>+</**a**> </**li**> |

cart.html实现删除功能(117行)：

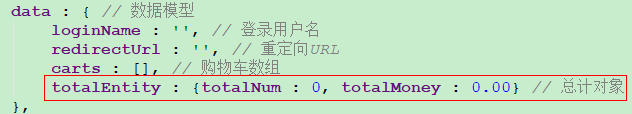
|  |
| --- |
| <**a href="javascript:;"  @click="addCart(orderItem.itemId,-orderItem.num)"**>删除</**a**><**br**/> |

## 4.4购买总数与购买总金额

### 4.4.1前端控制层

pinyougou-cart-web/src/main/webapp/js/controller/cartController.js文件：

* 在data中定义变量



* 修改methods中findCart方法：

|  |
| --- |
| *// 查询购物车* findCart : **function** () {  **axios**.get(**"/cart/findCart"**).then(**function**(response){  vue.**carts** = response.**data**;  */\*\* 清空总计对象 \*/* vue.**totalEntity** = {**totalNum** : 0, **totalMoney** : 0.00};  **for** (**var** i = 0; i < response.**data**.**length**; i++){  *// 获取购物车* **var** cart = response.**data**[i];  *// 迭代购物车订单明细集合* **for** (**var** j = 0; j < cart.**orderItems**.length; j++){  *// 获取订单明细* **var** orderItem = cart.**orderItems**[j];  *// 购买总件数* vue.**totalEntity**.**totalNum** += orderItem.num;  *// 购买总金额* vue.**totalEntity**.**totalMoney** += orderItem.totalFee;  }  }  }); }, |

### 4.4.2页面

修改cart.html(138行)：

|  |
| --- |
| <**div class="chosed"**>已选择  <**b**>{{**totalEntity**.**totalNum**}}</**b**>件商品 </**div**> <**div class="sumprice"**>  <**span**><**em**>总价（不含运费） ：</**em**>  <**i class="summoney"**>¥{{**totalEntity**.**totalMoney**.toFixed(2)}}</**i**>  </**span**> </**div**> |

# 5.Redis存储购物车【操作Redis】

## 5.1需求分析

判断当前用户是否登录，如果未登录采用Cookie存储，如果登录则采用Redis存储。登录后要进行Cookie购物车与Redis购物车的合并操作，并清除Cookie购物车。

说明：Redis的value存储List<Cart>二进制数据。

## 5.2购物车存取

### 5.2.1控制层

pinyougou-cart-web\src\main\java\com\pinyougou\cart\controller\CartController.java类，修改findCart方法：

|  |
| --- |
| */\*\* 获取购物车集合 \*/* @GetMapping(**"/findCart"**) **public** List<Cart> findCart() {  *// 获取登录用户名* String username = **request**.getRemoteUser();  *// 定义购物车集合* List<Cart> carts = **null**;  *// 判断用户是否登录* **if** (StringUtils.*isNoneBlank*(username)){ *// 已登录  /\*\*######## 从Redis获取购物车 #######\*/* carts = **cartService**.findCartRedis(username);  }**else** { *// 未登录  /\*\*######## 从Cookie获取购物车 #######\*/  // 从Cookie中获取购物车集合json字符串* String cartStr = CookieUtils.*getCookieValue*(**request**,  CookieUtils.CookieName.***PINYOUGOU\_CART***, **true**);  *// 判断是否为空* **if** (StringUtils.*isBlank*(cartStr)) {  cartStr = **"[]"**;  }  carts = JSON.*parseArray*(cartStr, Cart.**class**);  }  **return** carts; } |

pinyougou-cart-web\src\main\java\com\pinyougou\cart\controller\CartController.java类，修改addCart方法：

|  |
| --- |
| */\*\* 添加SKU商品到购物车 \*/* @GetMapping(**"/addCart"**) **public boolean** addCart(Long itemId, Integer num){  **try**{  *// 获取登录用户名* String username = **request**.getRemoteUser();  *// 获取购物车集合* List<Cart> carts = findCart();  *// 调用服务层添加SKU商品到购物车* carts = **cartService**.addItemToCart(carts, itemId, num);  *// 判断用户是否登录* **if** (StringUtils.*isNoneBlank*(username)){ *// 已登录  /\*\*######## 往Redis存储购物车 #######\*/* **cartService**.saveCartRedis(username, carts);  }**else**{ *// 未登录  /\*\*######## 往Cookie存储购物车 #######\*/  // 将购物车重新存入Cookie中* CookieUtils.*setCookie*(**request**, **response**,  CookieUtils.CookieName.***PINYOUGOU\_CART***,  JSON.*toJSONString*(carts),  3600 \* 24, **true**);  }  **return true**;  }**catch** (Exception ex){  ex.printStackTrace();  }  **return false**; } |

### 5.2.2服务接口层

pinyougou-interface\src\main\java\com\pinyougou\service\CartService.java新增方法：

|  |
| --- |
| */\*\*  \* 将购物车保存到Redis  \** ***@param username*** *用户名  \** ***@param carts*** *购物车  \*/* **void** saveCartRedis(String username, List<Cart> carts);  */\*\*  \* 从Redis中查询购物车  \** ***@param username*** *用户名  \** ***@return*** *购物车  \*/* List<Cart> findCartRedis(String username); |

### 5.2.3服务实现层

pinyougou-cart-service/src/main/java/com/pinyougou/cart/service/impl/CartServiceImpl.java实现方法：

|  |
| --- |
| @Autowired **private** RedisTemplate<String, Object> **redisTemplate**; */\*\*  \* 从Redis中查询购物车  \** ***@param username*** *用户名  \** ***@return*** *购物车  \*/* **public** List<Cart> findCartRedis(String username){  System.***out***.println(**"获取Redis中购物车: "** + username);  List<Cart> carts = (List<Cart>)**redisTemplate** .boundValueOps(**"cart\_"** + username).get();  **if**(carts == **null**){  carts = **new** ArrayList<>(0);  }  **return** carts; } */\*\*  \* 将购物车保存到Redis  \** ***@param username*** *用户名  \** ***@param carts*** *购物车  \*/* **public void** saveCartRedis(String username, List<Cart> carts){  System.***out***.println(**"往Redis存入购物车: "** + username);  **redisTemplate**.boundValueOps(**"cart\_"** + username).set(carts); } |

## 5.3购物车合并

### 5.3.1控制层

pinyougou-cart-web\src\main\java\com\pinyougou\cart\controller\CartController.java类，修改findCart方法：

|  |
| --- |
| */\*\* 获取购物车集合 \*/* @GetMapping(**"/findCart"**) **public** List<Cart> findCart() {  *// 获取登录用户名* String username = **request**.getRemoteUser();  *// 定义购物车集合* List<Cart> carts = **null**;  *// 判断用户是否登录* **if** (StringUtils.*isNoneBlank*(username)){ *// 已登录  /\*\*######## 从Redis获取购物车 #######\*/* carts = **cartService**.findCartRedis(username);  *// 从Cookie中获取购物车集合json字符串* String cartStr = CookieUtils.*getCookieValue*(**request**,  CookieUtils.CookieName.***PINYOUGOU\_CART***, **true**);  *// 判断是否为空* **if** (StringUtils.*isNoneBlank*(cartStr)) {  *// 转化成List集合* List<Cart> cookieCarts = JSON.*parseArray*(cartStr, Cart.**class**);  **if** (cookieCarts != **null** && cookieCarts.size() > 0) {  *// 合并购物车* carts = **cartService**.mergeCart(cookieCarts, carts);  *// 将合并后的购物车存入Redis* **cartService**.saveCartRedis(username, carts);  *// 删除Cookie购物车* CookieUtils.*deleteCookie*(**request**, **response**,  CookieUtils.CookieName.***PINYOUGOU\_CART***);  }  }  }**else** { *// 未登录  /\*\*######## 从Cookie获取购物车 #######\*/  // 从Cookie中获取购物车集合json字符串* String cartStr = CookieUtils.*getCookieValue*(**request**,  CookieUtils.CookieName.***PINYOUGOU\_CART***, **true**);  *// 判断是否为空* **if** (StringUtils.*isBlank*(cartStr)) {  cartStr = **"[]"**;  }  carts = JSON.*parseArray*(cartStr, Cart.**class**);  }  **return** carts; } |

### 5.3.2服务接口层

pinyougou-interface\src\main\java\com\pinyougou\service\CartService.java，新增方法：

|  |
| --- |
| */\*\*  \* 合并购物车  \** ***@param cookieCarts*** *Cookie购物车  \** ***@param redisCarts*** *Redis购物车  \** ***@return*** *合并后的购物车  \*/* List<Cart> mergeCart(List<Cart> cookieCarts, List<Cart> redisCarts); |

### 5.3.3服务实现层

pinyougou-cart-service/src/main/java/com/pinyougou/cart/service/impl/CartServiceImpl.java类中，实现方法：

|  |
| --- |
| */\*\* 合并购物车 \*/* **public** List<Cart> mergeCart(List<Cart> cookieCarts, List<Cart> redisCarts){  *// 迭代Cookie中的购物车数据* **for** (Cart cart : cookieCarts){  *// 迭代订单明细* **for** (OrderItem orderItem : cart.getOrderItems()){  redisCarts = addItemToCart(redisCarts,  orderItem.getItemId(), orderItem.getNum());  }  }  **return** redisCarts; } |

# 6.门户系统集成CAS

## 6.1配置集成CAS

pinyougou-portal-web/pom.xml：

|  |
| --- |
| <**dependencies**>  *<!-- cas-client-core -->* <**dependency**>  <**groupId**>org.jasig.cas.client</**groupId**>  <**artifactId**>cas-client-core</**artifactId**>  </**dependency**> </**dependencies**> |

pinyougou-portal-web/src/main/webapp/WEB-INF/web.xml：

|  |
| --- |
| *<!--############## CAS单点登录配置 开始 ################-->  <!-- 配置单点退出过滤器 (可选配置) -->* <**filter**>  <**filter-name**>ssoutFilter</**filter-name**> <**filter-class**>org.jasig.cas.client.session.SingleSignOutFilter</**filter-class**>  *<!-- 配置CAS服务URL前缀 -->* <**init-param**>  <**param-name**>casServerUrlPrefix</**param-name**>  <**param-value**>http://sso.pinyougou.com</**param-value**>  </**init-param**>  </**filter**>  <**filter-mapping**>  <**filter-name**>ssoutFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**>  </**filter-mapping**>  *<!-- 配置身份认证过滤器 (不需要跳转到CAS登录系统) -->* <**filter**>  <**filter-name**>authenticationFilter</**filter-name**> <**filter-class**>org.jasig.cas.client.authentication.AuthenticationFilter</**filter-class**>  <**init-param**>  *<!-- 配置CAS服务端登录地址 -->* <**param-name**>casServerLoginUrl</**param-name**>  <**param-value**>http://sso.pinyougou.com</**param-value**>  </**init-param**>  <**init-param**>  *<!-- 配置服务名称 -->* <**param-name**>serverName</**param-name**>  <**param-value**>**http://www.pinyougou.com**</**param-value**>  </**init-param**>  *<!-- 配置gateway=true”时，启用异步方式认证 -->* <**init-param**>  <**param-name**>gateway</**param-name**>  <**param-value**>true</**param-value**>  </**init-param**>  </**filter**>  <**filter-mapping**>  <**filter-name**>authenticationFilter</**filter-name**>  <**url-pattern**>\*.html</**url-pattern**>  </**filter-mapping**>   *<!-- 该过滤器负责对ticket的校验工作(必须启用) -->* <**filter**>  <**filter-name**>ticketValidationFilter</**filter-name**> <**filter-class**>org.jasig.cas.client.validation.Cas20ProxyReceivingTicketValidationFilter</**filter-class**>  *<!-- 配置CAS服务URL前缀 -->* <**init-param**>  <**param-name**>casServerUrlPrefix</**param-name**>  <**param-value**>http://sso.pinyougou.com</**param-value**>  </**init-param**>  *<!-- 配置服务名称 -->* <**init-param**>  <**param-name**>serverName</**param-name**>  <**param-value**>**http://www.pinyougou.com**</**param-value**>  </**init-param**>  </**filter**>  <**filter-mapping**>  <**filter-name**>ticketValidationFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**>  </**filter-mapping**>  *<!--  配置HttpServletRequest请求包裹过滤器 (可选配置)  说明：通过HttpServletRequest的getRemoteUser()方法获得SSO登录用户名 -->* <**filter**>  <**filter-name**>httpServletRequestWrapperFilter</**filter-name**> <**filter-class**>org.jasig.cas.client.util.HttpServletRequestWrapperFilter</**filter-class**>  </**filter**>  <**filter-mapping**>  <**filter-name**>httpServletRequestWrapperFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**>  </**filter-mapping**>  *<!--############## CAS单点登录配置 结束 ################-->* |

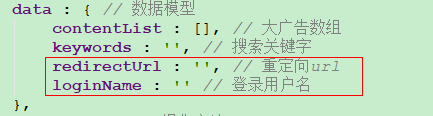
## 6.2登录用户名与退出

pinyougou-portal-web\src\main\java\com\pinyougou\portal\controller\LoginController.java：

|  |
| --- |
| @RestController **public class** LoginController {  */\*\* 显示用户名 \*/*  @GetMapping(**"/user/showName"**)  **public** Map<String,String> showName(HttpServletRequest request){  */\*\* 获取登录用户名 \*/* String name = request.getRemoteUser();  Map<String, String> map = **new** HashMap<>();  map.put(**"loginName"**, name);  **return** map;  } } |

pinyougou-portal-web\src\main\webapp\js\controller\indexController.js文件:

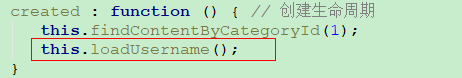
* 在data中定义变量



* 在methods中定义方法

|  |
| --- |
| *// 获取登录用户名* loadUsername : **function** () {  *// 定义重定向URL* **this**.**redirectUrl** = ***window***.*encodeURIComponent*(***location***.href);  *// 获取登录用户名* **axios**.get(**"/user/showName"**).then(**function**(response){  vue.**loginName** = response.**data**.**loginName**;  }); } |

* 初始化调用



pinyougou-portal-web/src/main/webapp/index.html页面：

|  |
| --- |
| <**ul class="fl"**>  <**li class="f-item" v-if="loginName"**>  {{**loginName**}}，品优购欢迎您！**&nbsp;** <**a :href="'http://sso.pinyougou.com/logout?service='+redirectUrl"**>退出</**a**>  </**li**>  <**li class="f-item" v-if="!loginName"**>请  <**a :href="'http://sso.pinyougou.com/?service='+redirectUrl"**>登录</**a**>  <**a href="http://user.pinyougou.com/register.html"**>免费注册</**a**>  </**li**> </**ul**> |

# 7.搜索系统集成CAS（学员挑战）

## 7.1配置集成CAS

pinyougou-search-web/pom.xml：

|  |
| --- |
| *<!-- cas-client-core -->* <**dependency**>  <**groupId**>org.jasig.cas.client</**groupId**>  <**artifactId**>cas-client-core</**artifactId**> </**dependency**> |

pinyougou-search-web/src/main/webapp/WEB-INF/web.xml：

|  |
| --- |
| *<!--############## CAS单点登录配置 开始 ################--> <!-- 配置单点退出过滤器 (可选配置) -->* <**filter**>  <**filter-name**>ssoutFilter</**filter-name**> <**filter-class**>org.jasig.cas.client.session.SingleSignOutFilter</**filter-class**>  *<!-- 配置CAS服务URL前缀 -->* <**init-param**>  <**param-name**>casServerUrlPrefix</**param-name**>  <**param-value**>http://sso.pinyougou.com</**param-value**>  </**init-param**> </**filter**> <**filter-mapping**>  <**filter-name**>ssoutFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**> </**filter-mapping**>  *<!-- 配置身份认证过滤器 (不需要跳转到CAS登录系统) -->* <**filter**>  <**filter-name**>authenticationFilter</**filter-name**> <**filter-class**>org.jasig.cas.client.authentication.AuthenticationFilter</**filter-class**>  <**init-param**>  *<!-- 配置CAS服务端登录地址 -->* <**param-name**>casServerLoginUrl</**param-name**>  <**param-value**>http://sso.pinyougou.com</**param-value**>  </**init-param**>  <**init-param**>  *<!-- 配置服务名称 -->* <**param-name**>serverName</**param-name**>  <**param-value**>http://search.pinyougou.com</**param-value**>  </**init-param**>  *<!-- 配置gateway=true”时，启用异步方式认证 -->* <**init-param**>  <**param-name**>gateway</**param-name**>  <**param-value**>true</**param-value**>  </**init-param**> </**filter**> <**filter-mapping**>  <**filter-name**>authenticationFilter</**filter-name**>  <**url-pattern**>\*.html</**url-pattern**> </**filter-mapping**>  *<!-- 该过滤器负责对ticket的校验工作(必须启用) -->* <**filter**>  <**filter-name**>ticketValidationFilter</**filter-name**> <**filter-class**>org.jasig.cas.client.validation.Cas20ProxyReceivingTicketValidationFilter</**filter-class**>  *<!-- 配置CAS服务URL前缀 -->* <**init-param**>  <**param-name**>casServerUrlPrefix</**param-name**>  <**param-value**>http://sso.pinyougou.com</**param-value**>  </**init-param**>  *<!-- 配置服务名称 -->* <**init-param**>  <**param-name**>serverName</**param-name**>  <**param-value**>http://search.pinyougou.com</**param-value**>  </**init-param**> </**filter**> <**filter-mapping**>  <**filter-name**>ticketValidationFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**> </**filter-mapping**>  *<!--  配置HttpServletRequest请求包裹过滤器 (可选配置)  说明：通过HttpServletRequest的getRemoteUser()方法获得SSO登录用户名 -->* <**filter**>  <**filter-name**>httpServletRequestWrapperFilter</**filter-name**> <**filter-class**>org.jasig.cas.client.util.HttpServletRequestWrapperFilter</**filter-class**> </**filter**> <**filter-mapping**>  <**filter-name**>httpServletRequestWrapperFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**> </**filter-mapping**> *<!--############## CAS单点登录配置 结束 ################-->* |

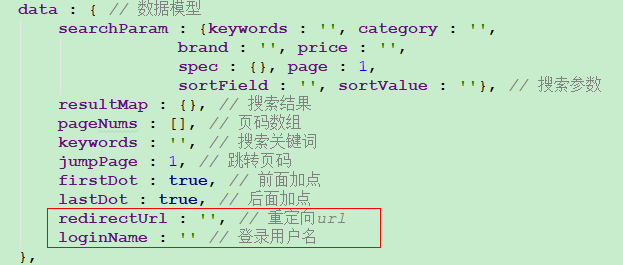
## 7.2登录用户名与退出

pinyougou-search-web\src\main\java\com\pinyougou\search\controller\LoginController.java：

|  |
| --- |
| @RestController **public class** LoginController {  */\*\* 显示用户名 \*/*  @GetMapping(**"/user/showName"**)  **public** Map<String,String> showName(HttpServletRequest request){  */\*\* 获取登录用户名 \*/* String name = request.getRemoteUser();  Map<String, String> map = **new** HashMap<>();  map.put(**"loginName"**, name);  **return** map;  } } |

pinyougou-search-web\src\main\webapp\js\controller\searchController.js文件：

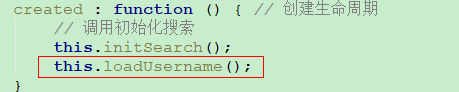
* 在data中定义变量



* 在methods中定义方法

|  |
| --- |
| *// 获取登录用户名* loadUsername : **function** () {  *// 定义重定向URL* **this**.**redirectUrl** = ***window***.*encodeURIComponent*(***location***.href);  *// 获取登录用户名* **axios**.get(**"/user/showName"**).then(**function**(response){  vue.**loginName** = response.**data**.**loginName**;  }); } |

* 初始化调用



pinyougou-search-web\src\main\webapp\index.html页面：

|  |
| --- |
| <**ul class="fl"**>  <**li class="f-item" v-if="loginName"**>  {{**loginName**}}，品优购欢迎您！**&nbsp;** <**a :href="'http://sso.pinyougou.com/logout?service='+redirectUrl"**>退出</**a**>  </**li**>  <**li class="f-item" v-if="!loginName"**>请  <**a :href="'http://sso.pinyougou.com/?service='+redirectUrl"**>登录</**a**>  <**a href="http://user.pinyougou.com/register.html"**>免费注册</**a**>  </**li**> </**ul**> |

# 8.详情系统集成CAS（学员挑战）

## 8.1配置集成CAS

pinyougou-item-web/pom.xml：

|  |
| --- |
| *<!-- jackson-databind -->* <**dependency**>  <**groupId**>com.fasterxml.jackson.core</**groupId**>  <**artifactId**>jackson-databind</**artifactId**> </**dependency**> *<!-- cas-client-core -->* <**dependency**>  <**groupId**>org.jasig.cas.client</**groupId**>  <**artifactId**>cas-client-core</**artifactId**> </**dependency**> |

pinyougou-item-web/src/main/webapp/WEB-INF/web.xml：

|  |
| --- |
| <!--  **因为获取登录名的请求URL为：/user/showName 不能进入SpringMVC的控制器。**  -->  <**servlet-mapping**>  <**servlet-name**>pinyougou</**servlet-name**>  <**url-pattern**>/</**url-pattern**> </**servlet-mapping**>  *<!--############## CAS单点登录配置 开始 ################--> <!-- 配置单点退出过滤器 (可选配置) -->* <**filter**>  <**filter-name**>ssoutFilter</**filter-name**> <**filter-class**>org.jasig.cas.client.session.SingleSignOutFilter</**filter-class**>  *<!-- 配置CAS服务URL前缀 -->* <**init-param**>  <**param-name**>casServerUrlPrefix</**param-name**>  <**param-value**>http://sso.pinyougou.com</**param-value**>  </**init-param**> </**filter**> <**filter-mapping**>  <**filter-name**>ssoutFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**> </**filter-mapping**>  *<!-- 配置身份认证过滤器(不需要跳转到CAS登录系统) -->* <**filter**>  <**filter-name**>authenticationFilter</**filter-name**> <**filter-class**>org.jasig.cas.client.authentication.AuthenticationFilter</**filter-class**>  <**init-param**>  *<!-- 配置CAS服务端登录地址 -->* <**param-name**>casServerLoginUrl</**param-name**>  <**param-value**>http://sso.pinyougou.com</**param-value**>  </**init-param**>  <**init-param**>  *<!-- 配置服务名称 -->* <**param-name**>serverName</**param-name**>  <**param-value**>http://item.pinyougou.com</**param-value**>  </**init-param**>  *<!-- 配置gateway为true，启动异步认证 -->* <**init-param**>  <**param-name**>gateway</**param-name**>  <**param-value**>true</**param-value**>  </**init-param**> </**filter**> <**filter-mapping**>  <**filter-name**>authenticationFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**> </**filter-mapping**>  *<!-- 该过滤器负责对ticket的校验工作(必须启用) -->* <**filter**>  <**filter-name**>ticketValidationFilter</**filter-name**> <**filter-class**>org.jasig.cas.client.validation.Cas20ProxyReceivingTicketValidationFilter</**filter-class**>  *<!-- 配置CAS服务URL前缀 -->* <**init-param**>  <**param-name**>casServerUrlPrefix</**param-name**>  <**param-value**>http://sso.pinyougou.com</**param-value**>  </**init-param**>  *<!-- 配置服务名称 -->* <**init-param**>  <**param-name**>serverName</**param-name**>  <**param-value**>http://item.pinyougou.com</**param-value**>  </**init-param**> </**filter**> <**filter-mapping**>  <**filter-name**>ticketValidationFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**> </**filter-mapping**>  *<!--  配置HttpServletRequest请求包裹过滤器 (可选配置)  说明：通过HttpServletRequest的getRemoteUser()方法获得SSO登录用户名 -->* <**filter**>  <**filter-name**>httpServletRequestWrapperFilter</**filter-name**> <**filter-class**>org.jasig.cas.client.util.HttpServletRequestWrapperFilter</**filter-class**> </**filter**> <**filter-mapping**>  <**filter-name**>httpServletRequestWrapperFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**> </**filter-mapping**> *<!--############## CAS单点登录配置 结束 ################-->* |

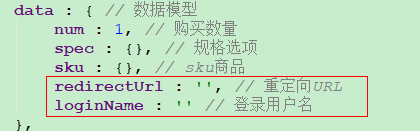
## 8.2登录用户名与退出

pinyougou-item-web\src\main\java\com\pinyougou\item\controller\LoginController.java：

|  |
| --- |
| @RestController **public class** LoginController {  */\*\* 显示用户名 \*/*  @GetMapping(**"/user/showName"**)  **public** Map<String,String> showName(HttpServletRequest request){  */\*\* 获取登录用户名 \*/* String name = request.getRemoteUser();  Map<String, String> map = **new** HashMap<>();  map.put(**"loginName"**, name);  **return** map;  } } |

pinyougou-search-web\src\main\webapp\js\controller\searchController.js文件：

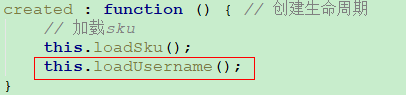
* 在data中定义变量



* 在methods中定义方法

|  |
| --- |
| *// 加载用户* loadUsername : **function** () {  *// 定义重定向URL* **this**.**redirectUrl** = ***window***.*encodeURIComponent*(***location***.href);  *// 获取登录用户名* **axios**.get(**"/user/showName"**).then(**function**(response){  vue.**loginName** = response.**data**.**loginName**;  }); } |

* 初始化调用



pinyougou-item-web\src\main\webapp\WEB-INF\ftl\header.ftl页面：

|  |
| --- |
| <**ul class="fl"**>  <**li class="f-item" v-if="loginName"**>  {{**loginName**}}，品优购欢迎您！**&nbsp;** <**a :href="'http://sso.pinyougou.com/logout?service='+redirectUrl"**>退出</**a**>  </**li**>  <**li class="f-item" v-if="!loginName"**>请  <**a :href="'http://sso.pinyougou.com/?service='+redirectUrl"**>登录</**a**>  <**a href="http://user.pinyougou.com/register.html"**>免费注册</**a**>  </**li**> </**ul**> |