**品优购电商系统开发**

**第19章**

**秒杀解决方案**

传智播客.黑马程序员

# 课程目标

目标1：秒杀实现思路

目标2：实现秒杀频道首页功能

目标3：实现秒杀商品详细页功能

目标4：实现秒杀下单功能

目标5：实现秒杀支付功能

# 秒杀业务分析

## 1.1需求分析

**什么是秒杀：**

秒杀场景一般会在电商网站举行一些活动。对于电商网站中一些稀缺或者特价商品，电商网站一般会在约定时间点对其进行限量销售，因为这些商品的特殊性，会吸引大量用户前来抢购，并且会在约定的时间点同时在秒杀页面进行抢购。由于商品价格低廉，往往一上架就被抢购一空，有时只用一秒钟。

**秒杀系统场景特点：**

* 秒杀时大量用户会在同一时间同时进行抢购，网站瞬时访问流量激增。
* 秒杀一般是访问请求数量远远大于库存数量，只有少部分用户能够秒杀成功。
* 秒杀业务流程比较简单，一般就是下订单减库存。

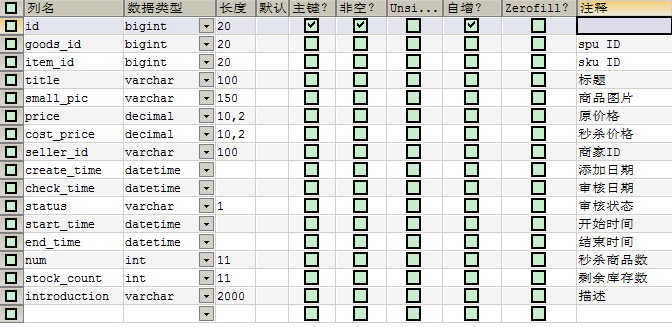
**秒杀商品通常有两种限制：库存限制、时间限制。**

**需求：**

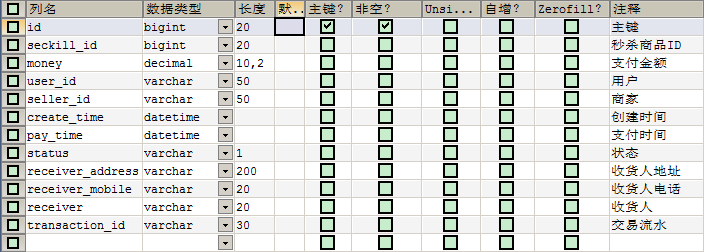
* 商家提交秒杀商品申请，录入秒杀商品数据，主要包括：商品标题、原价、秒杀价、商品图片、介绍等信息。
* 运营商审核秒杀申请。
* 秒杀频道首页列出秒杀商品（进行中的）点击秒杀商品图片跳转到秒杀商品详细页。
* 商品详细页显示秒杀商品信息，点击立即抢购实现秒杀下单，下单时扣减库存。当库存为0或不在活动期范围内时无法秒杀。
* 秒杀下单成功，直接跳转到支付页面（微信扫码），支付成功，跳转到成功页，填写收货地址、电话、收件人等信息，完成订单。
* 当用户秒杀下单5分钟内未支付，取消预订单，调用微信支付的关闭订单接口，恢复库存。

## 1.2数据库表分析

tb\_seckill\_goods秒杀商品表：



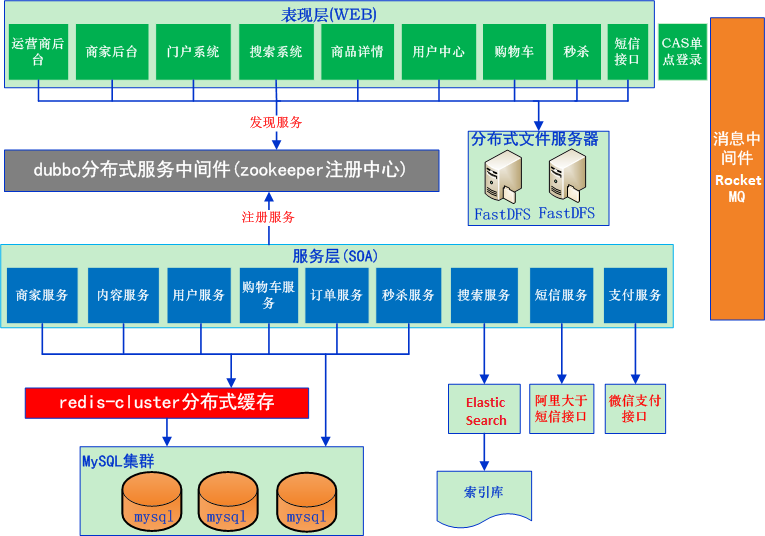
tb\_seckill\_order秒杀订单表：



## 1.3秒杀实现思路

秒杀技术实现核心思想是运用缓存减少数据库瞬间的访问压力！读取商品详细信息时运用缓存，当用户点击抢购时减少缓存中的库存数量，当库存数为0时或活动期结束时，同步到数据库。 产生的秒杀预订单也不会立刻写到数据库中，而是先写到缓存，当用户付款成功后再写入数据库。

# 搭建秒杀工程



## 2.1 pinyougou-seckill-service(war)

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

pinyougou-seckill-service/pom.xml：

|  |
| --- |
| <**properties**>  *<!-- 指定端口号 -->* <**port**>9009</**port**> </**properties**> |

pinyougou-seckill-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-seckill-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20888端口暴露服务 -->* <**dubbo:protocol name="dubbo" port="20888"**/>  *<!-- 配置采用包扫描来暴露服务 -->* <**dubbo:annotation package="com.pinyougou.seckill.service.impl"**/> </**beans**> |

## 2.2 pinyougou-seckill-web(war)

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

pinyougou-seckill-web/pom.xml：

|  |
| --- |
| <**properties**>  *<!-- 更改tomcat端口号 -->* <**port**>9110</**port**> </**properties**> <**dependencies**>  *<!-- cas-client-core -->* <**dependency**>  <**groupId**>org.jasig.cas.client</**groupId**>  <**artifactId**>cas-client-core</**artifactId**>  </**dependency**> </**dependencies**> |

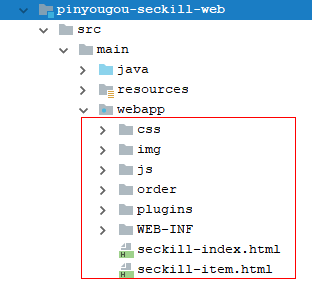
pinyougou-seckill-web/src/main/webapp/WEB-INF/web.xml：

|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* <**web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns="http://java.sun.com/xml/ns/javaee"  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee  http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"  version="3.0"**>  *<!-- 配置Spring MVC前端控制器(核心控制器) -->* <**servlet**>  <**servlet-name**>pinyougou</**servlet-name**> <**servlet-class**>org.springframework.web.servlet.DispatcherServlet</**servlet-class**>  <**init-param**>  <**param-name**>contextConfigLocation</**param-name**>  <**param-value**>classpath:springmvc.xml</**param-value**>  </**init-param**>  <**load-on-startup**>1</**load-on-startup**>  </**servlet**>  <**servlet-mapping**>  <**servlet-name**>pinyougou</**servlet-name**>  <**url-pattern**>/</**url-pattern**>  </**servlet-mapping**>   *<!-- 配置全局编码过滤器 -->* <**filter**>  <**filter-name**>characterEncodingFilter</**filter-name**> <**filter-class**>org.springframework.web.filter.CharacterEncodingFilter</**filter-class**>  <**init-param**>  <**param-name**>encoding</**param-name**>  <**param-value**>UTF-8</**param-value**>  </**init-param**>  </**filter**>  <**filter-mapping**>  <**filter-name**>characterEncodingFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**>  </**filter-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**>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://seckill.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://seckill.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://seckill.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 ################-->   <!-- 欢迎列表 -->* <**welcome-file-list**>  <**welcome-file**>seckill-index.html</**welcome-file**>  </**welcome-file-list**> </**web-app**> |

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

|  |
| --- |
| *<!-- 开启组件扫描 -->* <**context:component-scan base-package="com.pinyougou.seckill.controller"**/>  *......省略*  *<!-- ############## 配置dubbo服务消费者 ############## --> <!-- 配置当前应用的名称 -->* <**dubbo:application name="pinyougou-seckill-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.seckill.controller"**/> |

拷贝【资料\秒杀系统静态资源】目录下的文件到pinyougou-seckill-web/src/main/webapp/目录下。



## 2.3准备工作

### 2.3.1服务实现层

pinyougou-seckill-service/src/main/java/com/pinyougou/seckill/service.impl包下，创建SeckillGoodsServiceImpl.java服务接口实现：

|  |
| --- |
| @Service @Transactional **public class** SeckillGoodsServiceImpl **implements** SeckillGoodsService {  @Override  **public void** save(SeckillGoods seckillGoods) {  }  @Override  **public void** update(SeckillGoods seckillGoods) {  }  @Override  **public void** delete(Serializable id) {  }  @Override  **public void** deleteAll(Serializable[] ids) {  }  @Override  **public** SeckillGoods findOne(Serializable id) {  **return null**;  }  @Override  **public** List<SeckillGoods> findAll() {  **return null**;  }  @Override  **public** List<SeckillGoods> findByPage(SeckillGoods seckillGoods,  **int** page, **int** rows) {  **return null**;  } } |

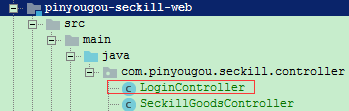
### 2.3.2控制器层

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

|  |
| --- |
| @RestController @RequestMapping(**"/seckill"**) **public class** SeckillGoodsController {  @Reference(timeout = 10000)  **private** SeckillGoodsService **seckillGoodsService**; } |

从pinyougou-cart-web/src/main/java/com/pinyougou/cart/controller/拷贝LoginController.java到pinyougou-seckill-web/src/main/java/

Com/pinyougou/seckill/controller/包下：



## 2.4配置域名访问

修改hosts文件：127.0.0.1 seckill.pinyougou.com

修改nginx.conf：

|  |
| --- |
| server {  listen 80;  server\_name seckill.pinyougou.com;  proxy\_set\_header X-Forwarded-Host $host;  proxy\_set\_header X-Forwarded-Server $host;  proxy\_set\_header Host $host;  location / {  proxy\_pass http://127.0.0.1:9110;  proxy\_connect\_timeout 600;  proxy\_read\_timeout 600;  }  } |

# 3.品优购【秒杀频道首页】

## 3.1需求分析

秒杀频道首页，显示正在秒杀的商品（已经开始，未结束的商品）

## 3.2前端代码

### 3.2.1控制器层

pinyougou-seckill-web\src\main\webapp\js\controller\seckillGoodsController.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* **seckillGoodsList** : [] *// 秒杀商品数组* },  **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**;  });  },  *// 查询秒杀商品* findSeckillGoods : **function** () {  **axios**.get(**"/seckill/findSeckillGoods"**).then(**function**(response){  vue.**seckillGoodsList** = response.**data**;  });  }  },  created : **function** () { *// 创建生命周期* **this**.loadUsername();  **this**.findSeckillGoods();  }  }); }; |

### 3.2.2页面

pinyougou-seckill-web/src/main/webapp/seckill-index.html，引入js：

|  |
| --- |
| <**script src="/js/vue.min.js"**></**script**> <**script src="/js/axios.min.js"**></**script**> <**script src="/js/controller/seckillGoodsController.js"**></**script**> |

seckill-index.html显示登录用户信息(26行)：

|  |
| --- |
| <**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**> |

seckill-index.html循环列表的实现(176行)：

|  |
| --- |
| <**li class="seckill-item" v-for="goods in seckillGoodsList"**>  <**div class="pic"**>  <**a :href="'/seckill-item.html?id=' + goods.id"**>  <**img :src="goods.smallPic" width="290" height="290"**/>  </**a**>  </**div**>  <**div class="intro"**>  <**span**>{{goods.title}}</**span**>  </**div**>  <**div class='price'**>  <**b class='sec-price'**>￥{{goods.costPrice}}</**b**>  <**b class='ever-price'**>￥{{goods.price}}</**b**>  </**div**>  <**div class='num'**>  <**div**>已售{{((goods.num-goods.stockCount)/goods.num\*100).toFixed(0)}}%</**div**>  <**div class='progress'**>  <**div class='sui-progress progress-danger'**>  <**span :style="'width:' + ((goods.num-goods.stockCount)/goods.num\*100).toFixed(0) +'%;'"  class='bar'**></**span**>  </**div**>  </**div**>  <**div**>剩余<**b class='owned'**>{{goods.stockCount}}</**b**>件</**div**>  </**div**>  <**a class='sui-btn btn-block btn-buy'  :href="'/seckill-item.html?id=' + goods.id"  target='\_blank'**>立即抢购</**a**> </**li**> |

## 3.3后端代码

### 3.3.1控制器层

pinyougou-seckill-web/src/main/java/com/pinyougou/seckill/controller/SeckillGoodsController.java：

|  |
| --- |
| */\*\* 查询秒杀商品列表 \*/* @GetMapping(**"/findSeckillGoods"**) **public** List<SeckillGoods> findSeckillGoods(){  **return seckillGoodsService**.findSeckillGoods(); } |

### 3.3.1服务接口层

pinyougou-interface\src\main\java\com\pinyougou\service\SeckillGoodsService.java：

|  |
| --- |
| */\*\* 查询秒杀的商品集合 \*/* List<SeckillGoods> findSeckillGoods(); |

### 3.3.2服务实现层

pinyougou-seckill-service\src\main\java\com\pinyougou\seckill\service\impl\SeckillGoodsServiceImpl.java：

|  |
| --- |
| @Autowired **private** SeckillGoodsMapper **seckillGoodsMapper**; */\*\* 查询秒杀的商品集合 \*/* **public** List<SeckillGoods> findSeckillGoods(){  **try**{  *// 创建示范对象* Example example = **new** Example(SeckillGoods.**class**);  *// 创建条件查询对象* Example.Criteria criteria = example.createCriteria();  *// 审核通过* criteria.andEqualTo(**"status"**, **"1"**);  *// 剩余库存大于0* criteria.andGreaterThan(**"stockCount"**, 0);  *// 开始时间小于等于当前时间* criteria.andLessThanOrEqualTo(**"startTime"**, **new** Date());  *// 结束时间大于等于当前时间* criteria.andGreaterThanOrEqualTo(**"endTime"**, **new** Date());  *// 条件查询*  **return seckillGoodsMapper**.selectByExample(example);  }**catch** (Exception ex){  **throw new** RuntimeException(ex);  } } |

## 3.4缓存处理

pinyougou-seckill-service\src\main\java\com\pinyougou\seckill\service\impl\SeckillGoodsServiceImpl.java的findSeckillGoods方法：

|  |
| --- |
| @Autowired **private** RedisTemplate **redisTemplate**; */\*\* 查询秒杀的商品集合 \*/* **public** List<SeckillGoods> findSeckillGoods(){  *// 定义秒杀商品数据* List<SeckillGoods> seckillGoodsList = **null**;  **try**{  *// 从Redis中获取秒杀商品数据* seckillGoodsList = **redisTemplate**.  boundHashOps(**"seckillGoodsList"**).values();  **if** (seckillGoodsList != **null** && seckillGoodsList.size() > 0){  System.***out***.println(**"Redis缓存中数据："** + seckillGoodsList);  **return** seckillGoodsList;  }  }**catch** (Exception ex){  ex.printStackTrace();  }  **try**{  */\*\* 创建示范对象 \*/* Example example = **new** Example(SeckillGoods.**class**);  */\*\* 创建条件查询对象 \*/* Example.Criteria criteria = example.createCriteria();  */\*\* 审核通过 \*/* criteria.andEqualTo(**"status"**, **"1"**);  */\*\* 剩余库存大于0 \*/* criteria.andGreaterThan(**"stockCount"**, 0);  */\*\* 开始时间小于等于当前时间 \*/* criteria.andLessThanOrEqualTo(**"startTime"**, **new** Date());  */\*\* 结束时间大于等于当前时间 \*/* criteria.andGreaterThanOrEqualTo(**"endTime"**, **new** Date());   *// 条件查询*  seckillGoodsList = **seckillGoodsMapper**.selectByExample(example);  System.***out***.println(**"==将秒杀商品存入缓存=="**);  **try**{  **for** (SeckillGoods seckillGoods : seckillGoodsList) {  **redisTemplate**.boundHashOps(**"seckillGoodsList"**)  .put(seckillGoods.getId(), seckillGoods);  }  }**catch** (Exception ex){  ex.printStackTrace();  }  **return** seckillGoodsList;  }**catch** (Exception ex){  **throw new** RuntimeException(ex);  } } |

# 4.品优购【秒杀详细页】

## 4.1需求分析

商品详细页显示秒杀商品详细信息。

## 4.2显示详细页信息

### 4.2.1后端代码

pinyougou-seckill-web\src\main\java\com\pinyougou\seckill\controller\SeckillGoodsController.java

|  |
| --- |
| */\*\* 根据秒杀商品id查询商品 \*/* @GetMapping(**"/findOne"**) **public** SeckillGoods findOne(Long id){  **return seckillGoodsService**.findOneFromRedis(id); } |

pinyougou-interface\src\main\java\com\pinyougou\service\SeckillGoodsService.java

|  |
| --- |
| */\*\* 根据秒杀商品id查询商品 \*/* SeckillGoods findOneFromRedis(Long id); |

pinyougou-seckill-service\src\main\java\com\pinyougou\seckill\service\impl\SeckillGoodsServiceImpl.java

|  |
| --- |
| */\*\* 根据秒杀商品id查询商品 \*/* **public** SeckillGoods findOneFromRedis(Long id){  **try**{  */\*\* 获取秒杀商品 \*/* **return** (SeckillGoods)**redisTemplate** .boundHashOps(**"seckillGoodsList"**).get(id);  }**catch** (Exception ex){  **throw new** RuntimeException(ex);  } } |

### 4.2.2前端代码

pinyougou-seckill-web\src\main\webapp\js\controller\seckillItemController.js:

|  |
| --- |
| *// 窗口加载完* $(**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* **entity** : {} *// 秒杀商品* },  **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**;  });  },  *// 根据秒杀商品id，查询秒杀商品* findOne : **function** () {  *// 获取请求URL后面的参数* **var** id = **this**.getUrlParam(**"id"**);  **axios**.get(**"/seckill/findOne?id="** + id).then(**function**(response){  vue.**entity** = response.**data**;  });  }  },  created : **function** () { *// 创建生命周期* **this**.loadUsername();  **this**.findOne();  }  }); }); |

pinyougou-seckill-web/src/main/webapp/seckill-item.html：

|  |
| --- |
| <**script src="/js/vue.min.js"**></**script**> <**script src="/js/axios.min.js"**></**script**> <**script src="/js/location.js"**></**script**> <**script src="/js/controller/seckillItemController.js"**></**script**> |

seckill-item.html显示登录用户信息(28行)：

|  |
| --- |
| <**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**> |

seckill-item.html用表达式显示标题(195行)：

|  |
| --- |
| <**div class="sku-name"**>  <**h4**>**{{entity**.**title}}**</**h4**> </**div**> |

seckill-item.html图片(186行)：

|  |
| --- |
| <**span class="jqzoom"**>  <**img :jqimg="entity.smallPic"  :src="entity.smallPic" width="400" height="400"**/> </**span**> |

seckill-item.html价格、剩余库存(202行)：

|  |
| --- |
| <**div class="summary-wrap"**>  <**div class="fl title"**>  <**i**>秒杀价</**i**>  </**div**>  <**div class="fl price"**><**i**>¥</**i**>  <**em**>{{**entity**.costPrice}}</**em**>  <**span**>原价：{{**entity**.price}}</**span**>  </**div**>  <**div class="fr remark"**>  剩余库存：{{**entity**.stockCount}}  </**div**> </**div**> |

seckill-item.html商品介绍(443行)：

|  |
| --- |
| <**div class="intro-detail"**>{{**entity**.introduction}}</**div**> |

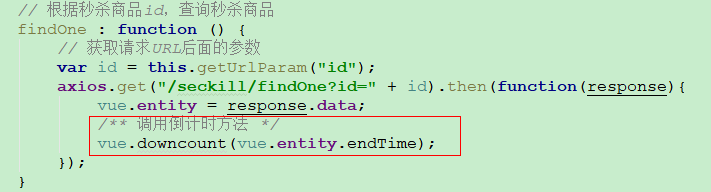
## 4.3秒杀倒计时效果

setTimeout延迟定时器，语法格式为：

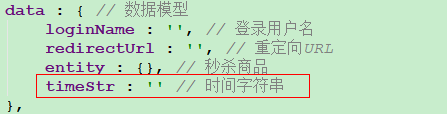
|  |
| --- |
| setTimeout(执行的函数,间隔的毫秒数); |

pinyougou-seckill-web\src\main\webapp\js\controller\seckillItemController.js文件：

* 修改methods中findOne方法



* 在data中定义变量



* 在methods中定义方法

|  |
| --- |
| *// 倒计时方法* downcount : **function** (endTime) {  */\*\* 计算出相差的毫秒数 \*/* **var** milliSeconds = endTime - **new *Date***().getTime();  */\*\* 计算出相差的秒数 \*/* **var** seconds = ***Math***.floor(milliSeconds / 1000);  */\*\* 判断秒是否大于零 \*/* **if** (seconds >= 0){  */\*\* 计算出分钟 \*/* **var** minutes = ***Math***.floor(seconds / 60);  */\*\* 计算出小时 \*/* **var** hours = ***Math***.floor(minutes / 60);  */\*\* 计算出天数 \*/* **var** days = ***Math***.floor(hours / 24);  */\*\* 定义resArr封装最后显示的时间 \*/* **var** resArr = **new *Array***();  **if** (days > 0){  resArr.push(**this**.calc(days) + **"天 "**);  }  **if** (hours > 0){  resArr.push(**this**.calc(hours - days \* 24) + **":"**);  }  **if** (minutes > 0){  resArr.push(**this**.calc(minutes - hours \* 60) + **":"**);  }  **if** (seconds >= 0){  resArr.push(**this**.calc(seconds - minutes \* 60));  }  **this**.**timeStr** = resArr.join(**""**);  */\*\* 开启延迟定时器 \*/  setTimeout*(**function**(){  vue.downcount(endTime);  }, 1000);  }**else**{  **this**.**timeStr** = **"秒杀结束！"**;  } }, *// 计算不够两位前面补零* calc : **function** (num) {  **return** num > 9 ? num : **"0"** + num; } |

pinyougou-seckill-web/src/main/webapp/seckill-item.html，显示倒计时的值(199行)

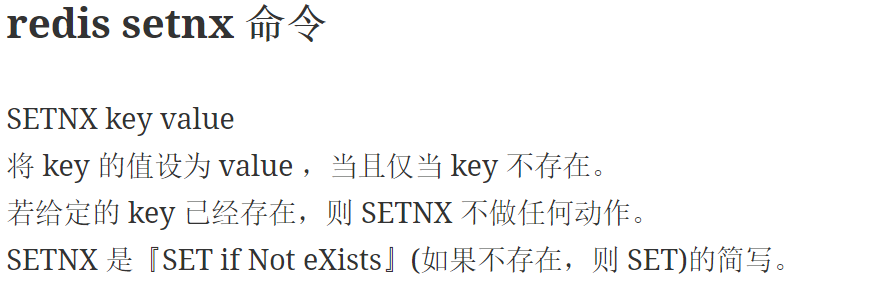
|  |
| --- |
| <**span class="overtime"**>距离结束：{{**timeStr**}}</**span**> |

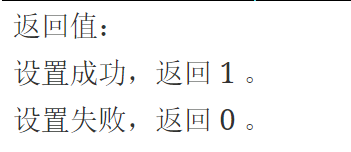
# 5.品优购【秒杀下单】

## 5.1需求分析

商品详细页点击立即抢购实现秒杀下单，订单先存储到Redis中减轻MySql数据库压力，下单时扣减库存(Redis中缓存商品的库存)。当库存为0时修改数据库秒杀商品库存同时删除Redis中秒杀商品。

对于分布式项目，我们可以使用Redis的setnx命令实现分布式锁，redis是单进程，天生就是线程安全。





## 5.2前端代码

### 5.2.1页面

pinyougou-seckill-web/src/main/webapp/seckill-item.html(253行)

|  |
| --- |
| <**a href="javascript:;" @click="submitOrder()"  class="sui-btn btn-danger addshopcar"**>立即抢购</**a**> |

### 5.2.2控制器层

pinyougou-seckill-web\src\main\webapp\js\controller\seckillItemController.js，在methods中定义方法：

|  |
| --- |
| *// 秒杀下单* submitOrder : **function** () {  *// 判断用户是否登录* **if** (**this**.**loginName**){ *// 已登录* **axios**.get(**"/order/submitOrder?id="** + **this**.**entity**.**id**)  .then(**function**(response){  **if** (response.**data**){  ***location***.href = **"/order/pay.html"**;  }**else** {  *alert*(**"下单失败！"**);  }  });  }**else**{ *// 未登录  // 跳转到单点登录系统* ***location***.href = **"http://sso.pinyougou.com?service="** + **this**.**redirectUrl**;  } } |

## 5.3后端代码

### 5.3.1控制器层

pinyougou-seckill-web\src\main\java\com\pinyougou\seckill\controller包下，创建SeckillOrderController.java

|  |
| --- |
| */\*\* 秒杀订单控制器 \*/* @RestController @RequestMapping(**"/order"**) **public class** SeckillOrderController {  @Reference(timeout = 10000)  **private** SeckillOrderService **seckillOrderService**;  */\*\* 提交订单到Redis \*/* @GetMapping(**"/submitOrder"**)  **public boolean** submitOrder(Long id,HttpServletRequest request){  **try**{  *// 获取登录用户名* String userId = request.getRemoteUser();  *// 提交订单到Redis* **seckillOrderService**.submitOrderToRedis(id, userId);  **return** true;  }**catch** (Exception ex){  ex.printStackTrace();  }  **return** false;  } } |

### 5.3.2服务接口层

pinyougou-interface\src\main\java\com\pinyougou\service\SeckillOrderService.java

|  |
| --- |
| */\*\*  \* 提交订单到Redis  \** ***@param id*** *秒杀商品id  \** ***@param userId*** *用户id  \*/* **void** submitOrderToRedis(Long id, String userId); |

### 5.3.3服务实现层

pinyougou-seckill-service/src/main/resources/applicationContext-service.xml，配置分布式id生成器

|  |
| --- |
| *<!-- 配置IdWorker分布式ID生成器 -->* <**bean id="idWorker" class="com.pinyougou.common.util.IdWorker"**>  *<!-- 工作id -->* <**constructor-arg name="workerId" value="1"**/>  *<!-- 数据中心id -->* <**constructor-arg name="datacenterId" value="0"**/> </**bean**> |

pinyougou-seckill-service\src\main\java\com\pinyougou\seckill\service\impl\包下，创建SeckillOrderServiceImpl.java

|  |
| --- |
| @Service(interfaceName = **"com.pinyougou.service.SeckillOrderService"**) @Transactional **public class** SeckillOrderServiceImpl **implements** SeckillOrderService {  @Autowired  **private** RedisTemplate **redisTemplate**;  @Autowired  **private** SeckillGoodsMapper **seckillGoodsMapper**;  @Autowired  **private** IdWorker **idWorker**;  */\*\*  \* 提交订单到Redis  \** ***@param id*** *秒杀商品id  \** ***@param userId*** *用户id  \*/* @Override  **public synchronized void** submitOrderToRedis(Long id, String userId){  **try**{   ***// 获取Redis分布式锁，如果为true代表获取锁成功* boolean lock = redisTemplate.opsForValue()**  **.setIfAbsent("kill\_" + id,true);**   **if (lock) { // 获取锁成功**  *// 从Redis中查询秒杀商品* SeckillGoods seckillGoods = (SeckillGoods) **redisTemplate**.  boundHashOps(**"seckillGoodsList"**).get(id);  *// 判断库存数据* **if** (seckillGoods != **null** && seckillGoods.getStockCount() > 0) {  *// 减库存(redis)* seckillGoods.setStockCount(seckillGoods.getStockCount() - 1);  *// 判断是否已经被秒光* **if** (seckillGoods.getStockCount() == 0) {  *// 同步秒杀商品到数据库(修改库存)* **seckillGoodsMapper**.updateByPrimaryKeySelective(seckillGoods);  *// 删除Redis中的秒杀商品* **redisTemplate**.boundHashOps(**"seckillGoodsList"**).delete(id);  } **else** {  *// 重新存入Redis中* **redisTemplate**.boundHashOps(**"seckillGoodsList"**)  .put(id, seckillGoods);  }  *// 创建秒杀订单对象* SeckillOrder seckillOrder = **new** SeckillOrder();  *// 设置订单id* seckillOrder.setId(**idWorker**.nextId());  *// 设置秒杀商品id* seckillOrder.setSeckillId(id);  *// 设置秒杀价格* seckillOrder.setMoney(seckillGoods.getCostPrice());  *// 设置用户id* seckillOrder.setUserId(userId);  *// 设置商家id* seckillOrder.setSellerId(seckillGoods.getSellerId());  *// 设置创建时间* seckillOrder.setCreateTime(**new** Date());  *// 设置状态码(未付款)* seckillOrder.setStatus(**"0"**);  *// 保存订单到Redis* **redisTemplate**.boundHashOps(**"seckillOrderList"**)  .put(userId, seckillOrder);  }  ***// 释放分布式锁* redisTemplate.delete("kill\_" + id);**  **}else{ // 获取锁失败**  **throw new** RuntimeException(**"该秒杀商品已锁！"**);  }  }**catch** (Exception ex){  ***// 释放分布式锁(怕出现死锁)* redisTemplate.delete("kill\_" + id);**  **throw new** RuntimeException(ex);  }  }  @Override  **public void** save(SeckillOrder seckillOrder) {  }  @Override  **public void** update(SeckillOrder seckillOrder) {  }  @Override  **public void** delete(Serializable id) {  }  @Override  **public void** deleteAll(Serializable[] ids) {  }  @Override  **public** SeckillOrder findOne(Serializable id) {  **return null**;  }  @Override  **public** List<SeckillOrder> findAll() {  **return null**;  }  @Override  **public** List<SeckillOrder> findByPage(SeckillOrder seckillOrder,  **int** page, **int** rows) {  **return null**;  } } |

# 6.品优购【秒杀支付】

## 6.1需求分析

用户成功下单后，跳转到支付页面。支付页显示微信支付二维码。用户完成支付后，保存订单到数据库。

## 6.2生成支付二维码

### 6.2.1前端代码

* seckillOrderController.js
* pay.html
* paysuccess.html
* payfail.html

说明：已准备好，不需要修改，具体参考购物车系统支付前端代码。

### 6.2.2后端代码

#### **6.2.2.1控制器层**

pinyougou-seckill-web\src\main\java\com\pinyougou\seckill\controller\SeckillOrderController.java

|  |
| --- |
| @Reference(timeout=10000) **private** WeixinPayService **weixinPayService**;  */\*\* 生成支付二维码 \*/* @GetMapping(**"/genPayCode"**) **public** Map<String, String> genPayCode(HttpServletRequest request){  */\*\* 获取登录用户名 \*/* String userId = request.getRemoteUser();  */\*\* 从Redis查询秒杀订单 \*/* SeckillOrder seckillOrder =  **seckillOrderService**.findOrderFromRedis(userId);   */\*\* 支付总金额（分）\*/* **long** totalFen = (**long**)(seckillOrder.getMoney().doubleValue() \* 100);  */\*\* 调用微信支付服务接口 \*/* **return weixinPayService**.genPayCode(  seckillOrder.getId().toString(),  String.*valueOf*(totalFen)); } |

#### **6.2.2.2服务接口层**

pinyougou-interface\src\main\java\com\pinyougou\service\SeckillOrderService.java

|  |
| --- |
| */\*\*  \* 根据用户名查询秒杀订单  \** ***@param userId*** *用户名  \*/* SeckillOrder findOrderFromRedis(String userId); |

#### **6.2.2.3服务实现层**

pinyougou-seckill-service\src\main\java\com\pinyougou\seckill\service\impl\SeckillOrderServiceImpl.java

|  |
| --- |
| */\*\* 根据用户名查询秒杀订单 \*/* **public** SeckillOrder findOrderFromRedis(String userId){  **try** {  *// 从Redis中查询用户秒杀订单* **return** (SeckillOrder)**redisTemplate** .boundHashOps(**"seckillOrderList"**).get(userId);  }**catch** (Exception ex){  **throw new** RuntimeException(ex);  } } |

## 6.3支付成功保存订单

### 6.3.1后端代码

#### **6.3.1.1控制器层**

pinyougou-seckill-web\src\main\java\com\pinyougou\seckill\controller\SeckillOrderController.java

|  |
| --- |
| */\*\* 查询支付状态 \*/* @GetMapping(**"/queryPayStatus"**) **public** Map<String,Integer> queryPayStatus(String outTradeNo,  HttpServletRequest request){  *// 定义Map集合封装数据* Map<String, Integer> data = **new** HashMap<>();  data.put(**"status"**, 3);  *// 调用查询订单接口* Map<String,String> map = **weixinPayService**.queryPayStatus(outTradeNo);  **if** (map != **null** && map.size() > 0){  *// 判断支付是否成功* **if** (**"SUCCESS"**.equals(map.get(**"trade\_state"**))){  *// 获取登录用户名* String userId = request.getRemoteUser();  *// 保存订单到数据库* **seckillOrderService**.saveOrder(userId,  map.get(**"transaction\_id"**));  data.put(**"status"**, 1);  }**else**{  data.put(**"status"**, 2);  }  }  **return** data; } |

#### **6.3.1.2服务接口层**

pinyougou-interface\src\main\java\com\pinyougou\service\SeckillOrderService.java

|  |
| --- |
| */\*\*  \* 支付成功保存订单  \** ***@param userId*** *用户名  \** ***@param transactionId*** *微信交易流水号  \*/* **void** saveOrder(String userId, String transactionId); |

#### **6.3.1.3服务实现层**

pinyougou-seckill-service\src\main\java\com\pinyougou\seckill\service\impl\SeckillOrderServiceImpl.java

|  |
| --- |
| @Autowired **private** SeckillOrderMapper **seckillOrderMapper**;  */\*\*  \* 支付成功保存订单  \** ***@param userId*** *用户名  \** ***@param transactionId*** *微信交易流水号  \*/* **public void** saveOrder(String userId, String transactionId){  **try**{  */\*\* 根据用户ID从redis中查询秒杀订单 \*/* SeckillOrder seckillOrder = (SeckillOrder)**redisTemplate** .boundHashOps(**"seckillOrderList"**).get(userId);  */\*\* 判断秒杀订单 \*/* **if**(seckillOrder != **null**){  */\*\* 微信交易流水号 \*/* seckillOrder.setTransactionId(transactionId);  */\*\* 支付时间 \*/* seckillOrder.setPayTime(**new** Date());  */\*\* 状态码(已付款) \*/* seckillOrder.setStatus(**"1"**);  */\*\* 保存到数据库 \*/* **seckillOrderMapper**.insertSelective(seckillOrder);  */\*\* 删除Redis中的订单 \*/* **redisTemplate.boundHashOps("seckillOrderList").delete(userId);**  }  }**catch** (Exception ex){  **throw new** RuntimeException(ex);  } } |