**新巴巴运动网 项目第一天**

**传智.上官云**

# 课前回顾

1. 去商品详情页面 加载（商品表、SKU表（包括颜色表） CMS项目中cmsService
2. SKU联动 （Jquery练习 ） 点击颜色 点击尺码 （页面初始化加载 触发器）
3. 静态化运行环境 1）实例化Configuration 2）设置模板所在目录 3）读取Template对象 4）Template.proccess(数据Map，Out)
4. 静态化标签：
5. el ${}
6. <#list list as l> </#list>
7. <#if l\_index == 0>ffff<#else>aaaa</if>
8. ${keyword!}
9. 容入项目中使用

Springmvc包中view/freemarker/freemarkerConfigrue.getConfigration 设置目录（相对）UTF-8

上架调用此静态化方法

1. 上架发送消息给MQ 接收方SOlr 接收方CMS 改主题：pubSubDomain true
2. CMS接收到商品ID　（Map）传递给静态化程序（生成静态页面）
3. 前台系统检索（每一个商品点击名称（http://localhost:8087/html/product/442.html）进入商品详情页面

# 今天内容

主打的是：Session共享 单点登陆

1、 单点登陆系统搭建

2、 去登陆页面 （提供成功后返回访问页面）

3、 提交登陆表单（面积消费者）用户体验

4、Session共享

5、 登陆成功后 跨域请求

6、购物车对象（表：面向对象思想创建一下购物车）

# 单点登陆系统搭建

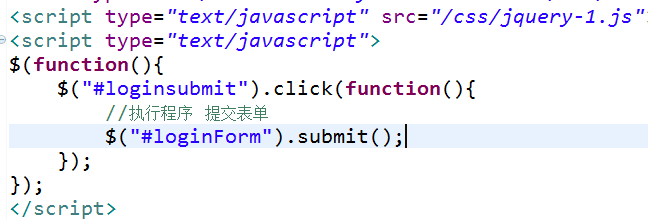
略

# 去登陆页面 （提供成功后返回访问页面）





# 提交登陆表单



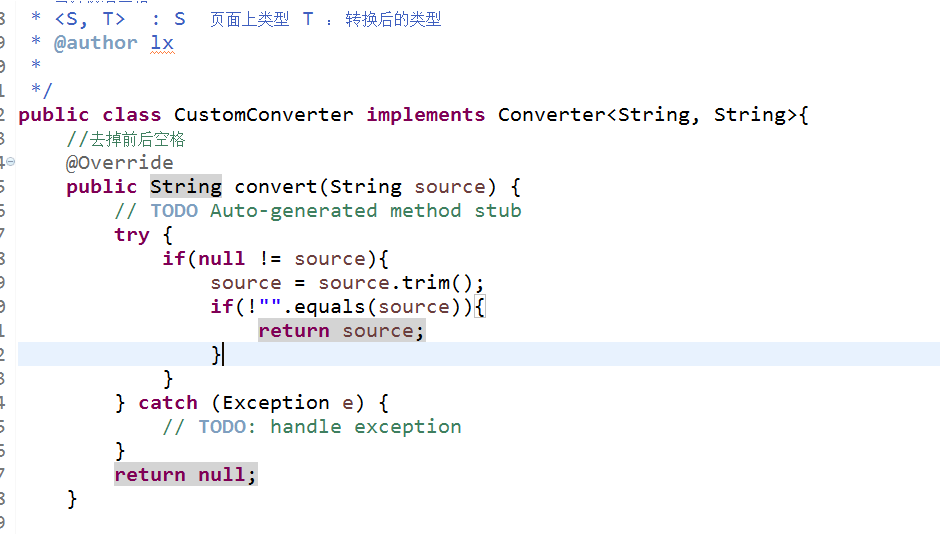
## 配置Springmvc的

Converter转换器

日期格式

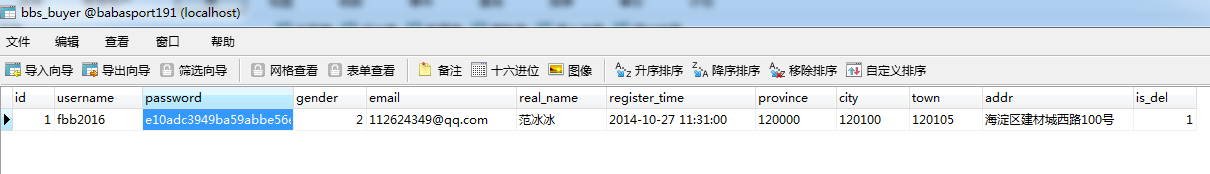
传递的参数（去掉前后格式） 如果还是空格转成Null 正常放行进行参数绑定

。。。





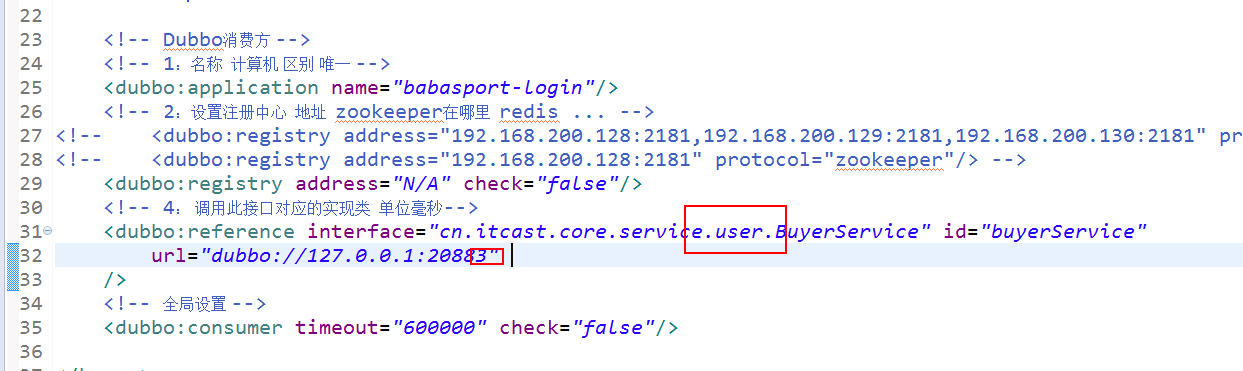
## 用户表



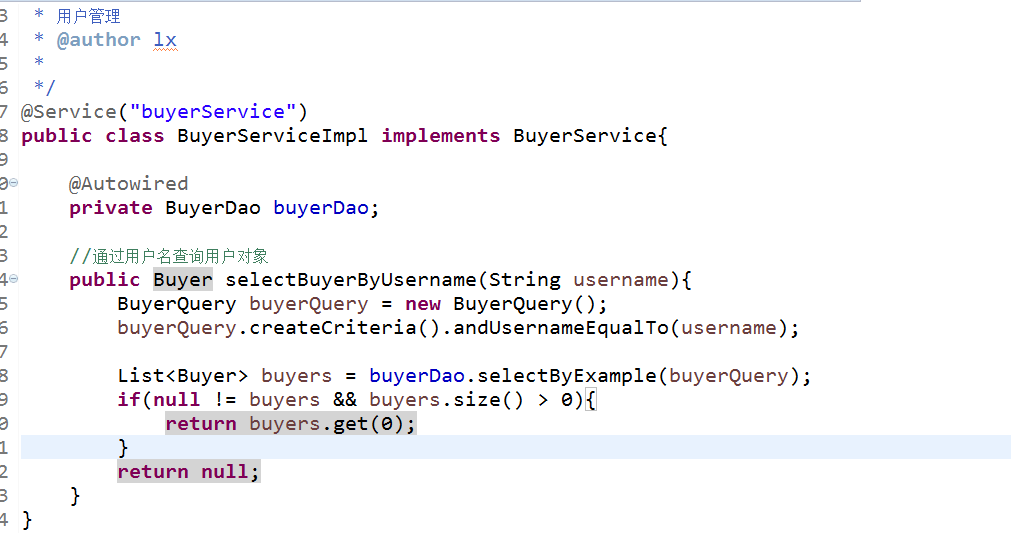
## Dubbo cob



## Dubbo login



## BuyerService



## 加密



## Session共享 为什么要使用Redis保存Session

系统 Tomcat 1

Request.session JSESSIONID

Redis服务器

K：V

负载均衡

用户

系统 Tomcat 2

系统 Tomcat 3

## Redis保存Session

本地Sesion

Request.getSesion.getAttr(Constants.USER\_NAME);

第一步：Cookie 中 JSESSIONID

K ： JSESSIONID

V ：Session

而Session

**K：**V

K：Constants.USER\_NAME

V: 用户对象（用户名）

K：验证码：名称

V：验证码的值

Redis中

K：V

　　　　K　：　JSESSIONID:Constants.USER\_NAME

　　　　V　：　用户名 　　　　（用户对象）用户对象转JSON字符串

验证码：

K ： JSESSIONID:Constants.CODE\_NAME

V ： 验证码值

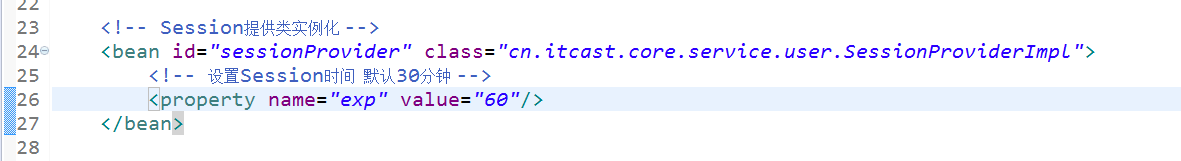
### 注意事项

Session过期时间 最后一次访问开始计时---30分钟

### SessionProvider （接口 实现类）

|  |
| --- |
| /\*\*  \* 保存用户名或验证码到Redis中  \* Session共享  \* @author lx  \*  \*/  public class SessionProviderImpl implements SessionProvider{  @Autowired  private Jedis jedis;  private Integer exp = 30;  public void setExp(Integer exp) {  this.exp = exp;  }  @Override  public void setAttribuerForUsername(String name, String value) {  // TODO Auto-generated method stub  //保存用户名到Redis中  //　K　：　CSESSIONID:Constants.USER\_NAME == name  jedis.set(name + ":" + Constants.USER\_NAME, value);  //时间  jedis.expire(name + ":" + Constants.USER\_NAME, 60\*exp);  }  @Override  public String getAttributeForUsername(String name) {  // TODO Auto-generated method stub  //fbb2016  String value = jedis.get(name + ":" + Constants.USER\_NAME);  if(null != value){  //时间  jedis.expire(name + ":" + Constants.USER\_NAME, 60\*exp);  }  return value;  }    } |

### 实例化Session



## Redis保存用户名设置时间

时间问题 ： 过期时间 30分钟

## 浏览器访问不同服务器时

JSESSIONID变化问题

CSESSIONID 自定义（随意起名）

工具类

方法：取CSESSIONID

生成CSESSIONID这件事

第一次问题CSESSIONID 自己创建一个UUID 36 去掉四个- 32位长度字符串 保存CSESSIONID到Cookie中

不是第一次：Cookie中有CSESSIONID 取出来使用

|  |
| --- |
| /\*\*  \* 获取CSESSIONID  \* @author lx  \*  \*/  public class RequestUtils {  //获取  public static String getCSESSIONID(HttpServletRequest request,HttpServletResponse response){  //1：取出Cookie  Cookie[] cookies = request.getCookies();  if(null != cookies && cookies.length > 0){  for (Cookie cookie : cookies) {  //2: 判断COokie中是否有CSESSIONID  if("CSESSIONID".equals(cookie.getName())){  //3:有 直接使用  return cookie.getValue();  }  }  }  //4:没有 创建一个CSESSIONID 并保存到COOKIE中 同时 把此COOKIe写回浏览器 使用此生成的CSESSIONID  String csessionid = UUID.randomUUID().toString().replaceAll("-", "");  Cookie cookie = new Cookie("CSESSIONID",csessionid);  //设置 存活时间 -1 0 >0  cookie.setMaxAge(-1);  //设置路径  cookie.setPath("/");  //设置跨域 localhost == www.babasport.com www.jd.com search.jd.com item.jd.com  //cookie.setDomain(".jd.com");  response.addCookie(cookie);  return csessionid;    }  } |

## LoginController

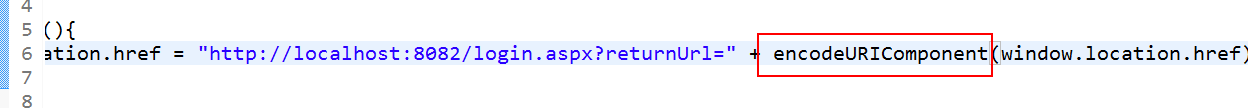
|  |
| --- |
| @Autowired  private BuyerService buyerService;  @Autowired  private SessionProvider sessionProvider;  //提交登陆  @RequestMapping(value = "/login.aspx",method=RequestMethod.POST)  public String login(String username,String password,String returnUrl,  HttpServletRequest request,HttpServletResponse response, Model model){  //1:用户名不能为空  if(null != username){  //2:密码不能为空  if(null != password){  //3:用户名必须正确  Buyer buyer = buyerService.selectBuyerByUsername(username);  if(null != buyer){  //4:密码必须正确  if(buyer.getPassword().equals(encodePassword(password))){  //5:保存用户名到Session中(Redis中）  sessionProvider.setAttribuerForUsername(RequestUtils.getCSESSIONID(request, response), buyer.getUsername());  //6:跳转到之前访问页面  return "redirect:" + returnUrl;  }else{  model.addAttribute("error", "密码必须正确");  }    }else{  model.addAttribute("error", "用户名必须正确");  }    }else{  model.addAttribute("error", "密码不能为空");  }    }else{  model.addAttribute("error", "用户名不能为空");  }  return "login";  } |

## 页面 login.jsp

回显错误



Login



# 页面

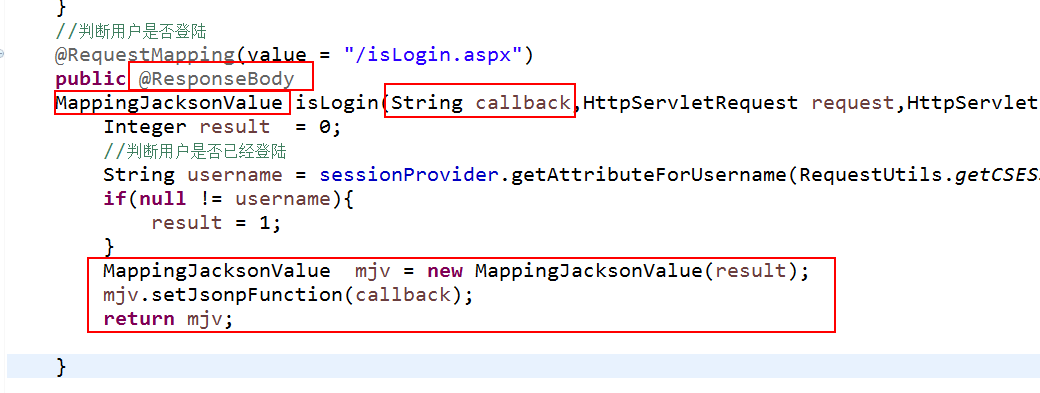


初始化加载Ajax 去取当前用户是否的登陆



## 跨源请求处理

LoginController



Shutcut.jsp

