



国际物流云商系统第十五天

一.回顾

- 1.消息中间件的介绍
- 2.消息中间件的使用场景及优势
- 3.两种消息模型
- 4.JMS 编程模型
- 5.ActiveMQ 的安装及使用
- 6.实现 Spring 与 ActiveMQ 整合开发

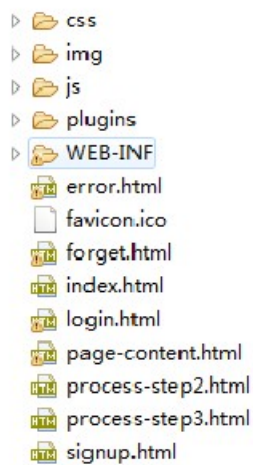
二. 国际物流云商系统的前端开发

该项目的前端系统将充分实现公司资源的整合，现阶段提供的功能主要包括一些基础功能，比如：前端普通客户的注册，提交提交，登录，密码找回功能，后期将加入电商系统的特点，实现前端基本项目特点。

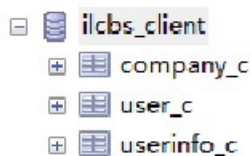
前端系统所需开发的模块包括

- ▷ bj44
- ▷ bj44_cache
- ▷ **bj44_client**
- ▷ bj44_exception
- ▷ bj44_jms
- ▷ bj44_rs
- ▷ bj44_server
- ▷ bj44_utils
- ▷ bj44_ws
- ▷ **bj44client_dao**
- ▷ **bj44client_domain**
- ▷ **bj44client_service**
- ▷ bj44rs_spring
- ▷ bj44server_dao
- ▷ bj44server_domain
- ▷ bj44server_service
- ▷ bj44ws_client
- ▷ bj44ws_spring

1. 添加前端系统开发的页面



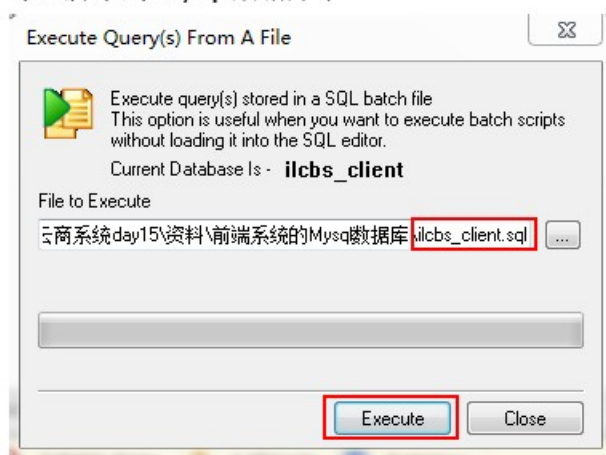
2. 添加前端系统的数据库表



将资料中的前端系统 sql 脚本导入到本地的 Mysql 数据库中。

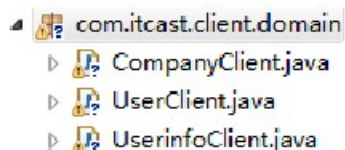


导入脚本到 Mysql 数据库中

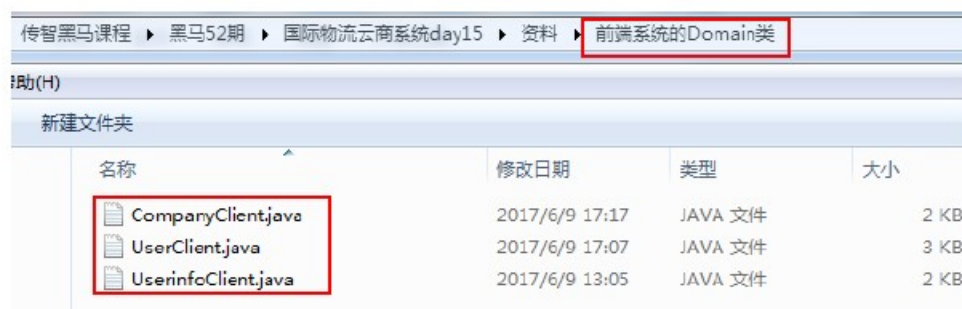




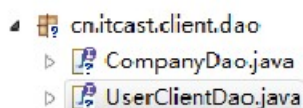
3. 添加前端系统的 Domain 实体类



资源位置



4. 添加前端系统的 Dao



接口声明如下:

```
public interface UserClientDao extends JpaRepository<UserClient, String> {

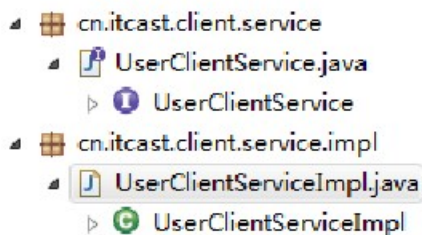
    UserClient findByUserNameAndPassword(String userName, String password);

    UserClient findByEmail(String email);
}
```

```
public interface CompanyDao extends JpaRepository<CompanyClient, String> {

}
```

5. 添加前端系统的 Service



业务接口中的方法声明



```
public interface UserClientService {
    //获取一条记录
    public UserClient get(Serializable id);
    //新增和修改保存
    public void saveOrUpdate(UserClient entity);

    //单条删除，按id
    public void deleteById(Serializable id);

    public UserClient findByUserNameAndPassword(String userName,String password);

    public UserClient findByEmail(String email);
}
```

6.添加前端系统的 Action

cn.itcast.client.action
BaseAction.java

三. 配置文件

1.添加 spring 相关配置文件

src/main/resources
applicationContext-mq.xml
applicationContext.xml
log4j.properties

2.添加并修改 web.xml 配置文件

Spring 配置文件加载

```
<context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>classpath:applicationContext.xml</param-value>
</context-param>
<listener>
    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>
```

为了解决 Spring data JPA 的 noSession 问题加入 OpenEntityManagerInViewFilter



```
<filter>
  <filter-name>openSessionInViewFilter</filter-name>
  <filter-class>org.springframework.orm.jpa.support.OpenEntityManagerInViewFilter</filter-class>
  <init-param>
    <param-name>entityManagerFactoryBeanName</param-name>
    <param-value>mysqlEntityManagerFactory</param-value>
  </init-param>
</filter>
<filter-mapping>
  <filter-name>openSessionInViewFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

Struts2 的核心控制器配置

```
<filter>
  <filter-name>struts2</filter-name>
  <filter-class>org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>struts2</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

3.运行效果展示



产品业务/Product Business

本次项目要开发注册功能，它所需的技术点

1. Jquery Validate 校验框架
2. 手机短信码的使用
3. Spring data Redis 实现短信码的缓存
4. 图片验证码的使用
5. ActiveMQ 消息队列的使用



四. 开发用户注册功能

1. 编写用户注册的 Action 类

```
public class UserAction extends BaseAction implements ModelDriven<UserClient>{  
  
    private UserClient model = new UserClient();  
    public UserClient getModel() {  
        return model;  
    }  
}
```

2. 注入 UserAction 开发的相关类

```
@Autowired  
@Qualifier("jmsQueueTemplate")  
private JmsTemplate jmsTemplate; // jms消息模板  
  
@Autowired  
private RedisTemplate<String, String> redisTemplate; // redis缓存模板  
  
@Autowired  
private UserService userClientService; // 用户业务逻辑
```

3. 生成图片验证码

```
@Action("userAction_genActiveCode")  
public String genActiveCode() throws Exception {  
    HttpServletResponse response = ServletActionContext.getResponse();  
    OutputStream out = response.getOutputStream(); // 得到输出流  
  
    String str = ImageUtil.getRandomStr();  
    System.out.println(str);  
  
    // 将生成的随机数放入HttpSession域中  
    session.put("vcode", str);  
  
    ImageUtil.getImage(str, out);  
    return NONE;  
}
```



4.手机验证码发送到消息系统

```
@Action(value="userAction_sendVerCode")
public String sendVerCode() throws Exception {
    //1.准备发送
    System.out.println(model.getTelephone());
    jmsTemplate.send("ilcbs_sms", new MessageCreator() {
        public Message createMessage(Session session) throws JMSException {
            MapMessage message = session.createMapMessage();

            message.setString("telephone", model.getTelephone());
            return message;
        }
    });
    return NONE;
}
```

消息系统处理短信消息

```
try {
    MapMessage msg = (MapMessage) message;
    String telephone = msg.getString("telephone");
    String mail = msg.getString("mail");
    // 如果传递的消息中有手机号，说明要发短信
    if (telephone != null) {
        String mobile_code = RandomCode.genCode()+"// 真正发送的短信验证码
        System.out.println("消费者QueueCustomer1,手机号:" + telephone + "验证码: "+mobile_code);

        // 发短信的服务
        String code = "2";
        AlibabaPhoneCodeGen.sendPhoneCode(telephone,mobile_code);

        // 保存短信信息到redis缓存中
        if ("2".equals(code)) {
            // 在redis缓存中保存了手机验证码 用户看到的是1分钟，但实质是3分钟
            redisTemplate.opsForValue().set(telephone, mobile_code, 3, TimeUnit.MINUTES);
        }
    }
}
```

5.用户注册方法编写

```
@Action(value="userAction_register")
public String register() throws Exception {
    String result="-1";
    //判断验证码，手机验证码是否正确
    //-----验证随机码
    String vcode = (String)session.get("vcode");
    if(!vcode.equals(vercode)){
        result="0";
        return NONE;
    }
    //-----验证手机验证码是否正确
    String telVerCode = redisTemplate.opsForValue().get(model.getTelephone());

    if(!telVerCode.equals(phoneVercode)){
        result="1";
        return NONE;
    }
}
```



```
//1.添加用户注册的信息
userService.saveOrUpdate(model);
//2.发送激活邮件
JmsTemplate.send("ilcbs_sms",new MessageCreator() {
    public Message createMessage(Session session) throws JMSException {
        MapMessage message = session.createMapMessage();

        message.setString("mail", model.getEmail());
        return message;
    }
});

result="2";//成功

//3.返回响应的结果
HttpServletResponse response = ServletActionContext.getResponse();
response.setContentType("text/html;charset=UTF-8");

response.getWriter().write(result);
return NONE;
```

6.注册成功时，消息系统实现邮件发送

```
else if (mail != null) {
    // 说明是发邮件
    String mail_code = "ilcbs_201706" + (int) ((Math.random() * 9 + 1) * 1000000) + "_199204";// 真正发送的短信验证码
    System.out.println("您的邮箱号: "+mail+", 邮箱激活码: "+mail_code);
    MailUtil.sendMsg(mail, "国际物流云商系统---用户注册的邮箱激活功能",
        "欢迎加入传智播客教育集团，您的激活链接是: <a href='\"http://localhost:8082/sz06/userAction_active?email="+mail+"&emailcode=" + mail_code+
        + "\"'/>,如果不能点击，请复制链接: http://localhost:8082/sz06/userAction_active?email=="+mail+"&emailcode=" + mail_code+

    // 同时将用户mail_code保存到redis缓存中
    redisTemplate.opsForValue().set(mail, mail_code, 3, TimeUnit.DAYS);//保存3天
```

五.项目实战需求分析

1.委托单的需求分析

共 10 分

1. 委托单的生成(新增)
状态为 0: 草稿
2. 委托单的查询
3. 委托单的修改&删除【可以修改状态也可以真实删除】
4. 委托单的提交与取消【状态的修改】

shopping_order		orderno (id)	city, fax
冠生 安泰		orderno (1)	
冠生 安泰		orderno (2)	
冠生 安泰		orderno (3)	
冠生 安泰		orderno (4)	
冠生 安泰		orderno (5)	
冠生 安泰		orderno (6)	
冠生 安泰		orderno (7)	
冠生 安泰		orderno (8)	
冠生 安泰		orderno (9)	
冠生 安泰		orderno (10)	
冠生 安泰		orderno (11)	
冠生 安泰		orderno (12)	
冠生 安泰		orderno (13)	
冠生 安泰		orderno (14)	
冠生 安泰		orderno (15)	
冠生 安泰		orderno (16)	
冠生 安泰		orderno (17)	
冠生 安泰		orderno (18)	
冠生 安泰		orderno (19)	
冠生 安泰		orderno (20)	

委托单新增

装箱单列表

要求，左边，右边表
单元素分开

委托单和装箱单形成一对一关系

委托单新增时，要求查询出装箱单列表（状态为1的），可以在显示的装箱单列表中，将前面的的选项设置为单选按钮。

委托单的id不设置为uuid主键生成方式，而是程序指定值，而程序指定的值来自于页面上的id单选按钮指定的id

2.发票单的需求分析

共计 10 分

1. 发票单的生成(新增)【状态 0】
2. 发票单的查询
3. 发票单的修改&删除
4. 发票单的提交与取消【提交时状态改为 1】

发票		
INVOICE_ID	varchar2(40)	(pk, fk)
SC_ID	varchar2(100)	
BL_ID	varchar2(100)	
发票条款	varchar2(100)	
状态	int	
创建人	varchar2(40)	
创建时间	varchar2(40)	
创建日期	datetime	

BL_NO:代表提单号,
谁有提单,谁就有提货权

发票新增

委托单列表：状态为1就显示

单选按钮

要求, 左边, 右边表
单元索分开

发票新增时，因为装箱单的发票号，前期是没有值的，所以当发票生成时，发票号就存在值了，就要更新装箱单的发票号

3.财务报运单需求分析

共计 10 分

1. 财务报运单的生成【新增】
2. 财务报运单的查询
3. 财务报运单的修改&删除
4. 财务报运单的提交与取消



4.基础信息分析

共计 25 分（包括 1,2,3 的功能）

1. 工厂信息的 CRUD
厂家删除时，只要修改状态为 2(代表这个厂家与我们公司没有合作了)
2. 产品信息的 CRUD
产品与工厂之间的关系（多对一）
开发时，可以在产品添加时，提供厂家下拉列表
3. 市场价最高的前 10 名产品（按市场价统计），并使用图形报表进行统计(JSON 数据)

共计 10 分

4. 登录日志的添加

```
//LoginLog log = new LoginLog();
//log.setUsername(user.getUsername());          log.setIpAddress(request.getRemoteAddr())
//加入日志的代码 loginLogService.saveOrUpdate();
```

- 5.
6. 登录 IP 次数的统计图，前 10 名（使用图形报表方式生成）[可选]

六.作业

- 1.回顾 Linux 常用命令（可以参考之前的视频或文档）
- 2.在 VMWare 中的 Linux 系统中，安装好 JDK
3. 在 VMWare 中的 Linux 系统中，安装 Tomcat
- 4.测试网络
- 5.实现项目部署到 Linux 中的 Tomcat
- 6.测试并访问项目