

1.1 CRM综合练习：客户拜访记录

1.1.1 客户拜访记录管理

1.1.1.1 客户拜访记录需求的概述

一个系统的用户（公司业务员），业务员需要对客户进行拜访，对拜访的过程进行记录。业务员（用户）和客户（客户）之间关系（需要具体业务具体分析：一对多—公司产品比较单一，只允许一个业务员对应多个客户。多对多—大公司有不同的产品，不同产品下有不同业务员都可以接触到同一个客户，一个客户可以对应多个业务员）。大部分情况创建成多对多。

- 多对多建表原则：
 - 创建中间表，中间表中至少两个字段分别作为外键指向多对多双方的主键。

1.1.2 客户拜访记录的准备工作

1.1.2.1 创建表

```
CREATE TABLE `sale_visit` (  
  `visit_id` varchar(32) NOT NULL,  
  `visit_cust_id` bigint(32) DEFAULT NULL COMMENT '客户id',  
  `visit_user_id` bigint(32) DEFAULT NULL COMMENT '负责人id',  
  `visit_time` datetime DEFAULT NULL COMMENT '拜访时间',  
  `visit_addr` varchar(128) DEFAULT NULL COMMENT '拜访地点',  
  `visit_detail` varchar(256) DEFAULT NULL COMMENT '拜访详情',  
  `visit_nexttime` date DEFAULT NULL COMMENT '下次拜访时间',  
  PRIMARY KEY (`visit_id`),  
  KEY `FK_sale_visit_cust_id` (`visit_cust_id`),  
  KEY `FK_sale_visit_user_id` (`visit_user_id`),  
  CONSTRAINT `FK_sale_visit_cust_id` FOREIGN KEY (`visit_cust_id`) REFERENCES  
  `cst_customer` (`cust_id`) ON DELETE NO ACTION ON UPDATE NO ACTION,  
  CONSTRAINT `FK_sale_visit_user_id` FOREIGN KEY (`visit_user_id`) REFERENCES  
  `sys_user` (`user_id`) ON DELETE NO ACTION ON UPDATE NO ACTION  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

1.1.2.2 创建实体和映射

- 创建实体

```
package com.admiral.crm.domain;  
  
import java.util.Date;  
  
/**  
 *  
 * @Description: 拜访记录的实体  
 * @author Admiral
```

```

*
*/
public class Salevisit {
    private String visit_id;
    private Date visit_time;
    private String visit_addr;
    private String visit_detail;
    private Date visit_nexttime;

    //拜访记录关联的客户对象
    private Customer customer;
    //拜访记录关联的用户对象
    private User user;
    public String getVisit_id() {
        return visit_id;
    }
    public void setVisit_id(String visit_id) {
        this.visit_id = visit_id;
    }
    public Date getVisit_time() {
        return visit_time;
    }
    public void setVisit_time(Date visit_time) {
        this.visit_time = visit_time;
    }
    public String getVisit_addr() {
        return visit_addr;
    }
    public void setVisit_addr(String visit_addr) {
        this.visit_addr = visit_addr;
    }
    public String getVisit_detail() {
        return visit_detail;
    }
    public void setVisit_detail(String visit_detail) {
        this.visit_detail = visit_detail;
    }
    public Date getVisit_nexttime() {
        return visit_nexttime;
    }
    public void setVisit_nexttime(Date visit_nexttime) {
        this.visit_nexttime = visit_nexttime;
    }
    public Customer getCustomer() {
        return customer;
    }
    public void setCustomer(Customer customer) {
        this.customer = customer;
    }
    public User getUser() {
        return user;
    }
    public void setUser(User user) {
        this.user = user;
    }
}

```

- 创建映射

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping PUBLIC
    "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
    "http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping>
    <class name="com.admiral.crm.domain.Salevisit" table="sale_visit">
        <id name="visit_id" column="visit_id">
            <generator class="uuid" />
        </id>

        <property name="visit_time" column="visit_time" />
        <property name="visit_addr" column="visit_addr" />
        <property name="visit_detail" column="visit_detail" />
        <property name="visit_nexttime" column="visit_nexttime" />

        <!-- 配置与客户的关联关系 -->
        <many-to-one name="customer" class="com.admiral.crm.domain.Customer"
column="visit_cust_id" />
        <!-- 配置与用户的关联关系 -->
        <many-to-one name="user" class="com.admiral.crm.domain.User"
column="visit_user_id" />

    </class>
</hibernate-mapping>
```

- 在 Spring 的配置文件中引入映射文件

```
<!-- 引入映射文件 -->
<property name="mappingResources">
    <list>
        <value>com/admiral/crm/domain/User.hbm.xml</value>
        <value>com/admiral/crm/domain/Customer.hbm.xml</value>
        <value>com/admiral/crm/domain/BaseDict.hbm.xml</value>
        <value>com/admiral/crm/domain/LinkMan.hbm.xml</value>
        <value>com/admiral/crm/domain/Salevisit.hbm.xml</value>
    </list>
</property>
```

1.1.2.3 创建相关类

- SaleVisitDao

```

package com.admiral.crm.dao;

import com.admiral.crm.domain.SaleVisit;

/**
 *
 * @Description: 拜访记录管理模块:Dao层的接口
 * @author Admiral
 */
public interface SaleVisitDao extends BaseDao<SaleVisit> {

}

```

- SaleVisitDaoImpl

```

package com.admiral.crm.dao.impl;

import com.admiral.crm.dao.SaleVisitDao;
import com.admiral.crm.domain.SaleVisit;

public class SaleVisitDaoImpl extends BaseDaoImpl<SaleVisit> implements
SaleVisitDao {

}

```

- SaleVisitService

```

package com.admiral.crm.service;

/**
 *
 * @Description: 客户拜访管理模块:Service层的接口
 * @author Admiral
 */
public interface SaleVisitService {

}

```

- SaleVisitServiceImpl

```

package com.admiral.crm.service.impl;

import com.admiral.crm.service.SaleVisitService;

public class SaleVisitServiceImpl implements SaleVisitService {

}

```

- SaleVisitAction

```

package com.admiral.crm.web.action;

import com.admiral.crm.domain.SaleVisit;
import com.opensymphony.xwork2.ActionSupport;
import com.opensymphony.xwork2.ModelDriven;

public class SaleVisitAction extends ActionSupport implements
ModelDriven<SaleVisit> {

    //模型驱动使用的对象
    private SaleVisit salevisit = new SaleVisit();

    @Override
    public SaleVisit getModel() {
        return salevisit;
    }

}

```

1.1.2.4 将相关类进行配置

- 开启属性注解

```

2
5      <!-- 配置在没有开启类扫描的情况下使用注解 -->
7      <context:annotation-config />
8  </beans>

```

- 在 Spring 中配置相关类

```

<?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:context="http://www.springframework.org/schema/context"
    xmlns:aop="http://www.springframework.org/schema/aop"
    xmlns:tx="http://www.springframework.org/schema/tx"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans.xsd
        http://www.springframework.org/schema/context
        http://www.springframework.org/schema/context/spring-context.xsd
        http://www.springframework.org/schema/aop
        http://www.springframework.org/schema/aop/spring-aop.xsd
        http://www.springframework.org/schema/tx
        http://www.springframework.org/schema/tx/spring-tx.xsd">

    <!-- ===== Spring 整合 Hibernate ===== -->

    <!-- 引入外部属性文件 -->
    <context:property-placeholder location="classpath:jdbc.properties"/>

    <!-- 配置 C3P0 连接池 -->

```

```

<bean id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource">
    <property name="driverClass" value="${jdbc.driverClass}" />
    <property name="jdbcUrl" value="${jdbc.url}" />
    <property name="user" value="${jdbc.username}" />
    <property name="password" value="${jdbc.password}" />
</bean>

<!-- 配置 Hibernate SessionFactory 相关配置 -->
<bean id="sessionFactory"
class="org.springframework.orm.hibernate5.LocalSessionFactoryBean">
    <!-- 注入连接池 -->
    <property name="dataSource" ref="dataSource" />
    <!-- 注入Hibernate 相关配置 -->
    <property name="hibernateProperties">
        <props>
            <prop
key="hibernate.dialect">org.hibernate.dialect.MySQLDialect</prop>
            <prop key="hibernate.show_sql">true</prop>
            <prop key="hibernate.format_sql">true</prop>
            <prop key="hibernate.hbm2ddl.auto">update</prop>
        </props>
    </property>
    <!-- 引入映射文件 -->
    <property name="mappingResources">
        <list>
            <value>com/admiral/crm/domain/User.hbm.xml</value>
            <value>com/admiral/crm/domain/Customer.hbm.xml</value>
            <value>com/admiral/crm/domain/BaseDict.hbm.xml</value>
            <value>com/admiral/crm/domain/LinkMan.hbm.xml</value>
            <value>com/admiral/crm/domain/SaleVisit.hbm.xml</value>
        </list>
    </property>
</bean>

<!-- ===== User 模块的相关配置 ===== -->
<!-- 配置 Action -->
<bean id="userAction" class="com.admiral.crm.web.action.UserAction"
scope="prototype">
    <property name="userService" ref="userService" />
</bean>
<!-- 配置 Service -->
<bean id="userService" class="com.admiral.crm.service.impl.UserServiceImpl">
    <property name="userDao" ref="userDao" />
</bean>
<!-- 配置 Dao -->
<bean id="userDao" class="com.admiral.crm.dao.impl.UserDaoImpl">
    <property name="sessionFactory" ref="sessionFactory" />
</bean>

<!-- ===== Customer 模块的相关配置 ===== -->
<bean id="customerDao" class="com.admiral.crm.dao.impl.CustomerDaoImpl">
    <property name="sessionFactory" ref="sessionFactory" />
</bean>
<bean id="customerService"
class="com.admiral.crm.service.impl.CustomerServiceImpl">
    <property name="customerDao" ref="customerDao" />

```

```

    </bean>
    <bean id="customerAction" class="com.admiral.crm.web.action.CustomerAction"
scope="prototype">
        <property name="customerService" ref="customerService" />
    </bean>

    <!-- ===== BaseDict 模块的相关配置 ===== -->
    <bean id="baseDictAction" class="com.admiral.crm.web.action.BaseDictAction"
scope="prototype">
        <property name="baseDictService" ref="baseDictService" />
    </bean>

    <bean id="baseDictService"
class="com.admiral.crm.service.impl.BaseDictServiceImpl">
        <property name="baseDictDao" ref="baseDictDao" />
    </bean>

    <bean id="baseDictDao" class="com.admiral.crm.dao.impl.BaseDictDaoImpl">
        <property name="sessionFactory" ref="sessionFactory" />
    </bean>

    <!-- ===== LinkMan 模块的相关配置 ===== -->
    <bean id="linkManAction" class="com.admiral.crm.web.action.LinkManAction"
scope="prototype">
        <property name="linkManService" ref="linkManService" />
        <property name="customerService" ref="customerService" />
    </bean>
    <bean id="linkManService"
class="com.admiral.crm.service.impl.LinkManServiceImpl">
        <property name="linkManDao" ref="linkManDao" />
    </bean>
    <bean id="linkManDao" class="com.admiral.crm.dao.impl.LinkManDaoImpl">
        <property name="sessionFactory" ref="sessionFactory" />
    </bean>

    <!-- ===== SaleVisit 模块的相关配置 ===== -->
    <bean id="saleVisitAction"
class="com.admiral.crm.web.action.SaleVisitAction" scope="prototype">

    </bean>

    <bean id="saleVisitService"
class="com.admiral.crm.service.impl.SaleVisitServiceImpl">

    </bean>

    <bean id="saleVisitDao" class="com.admiral.crm.dao.impl.SaleVisitDaoImpl">
        <property name="sessionFactory" ref="sessionFactory" />
    </bean>

    <!-- 配置平台事务管理器 -->
    <bean id="transactionManager"
class="org.springframework.orm.hibernate5.HibernateTransactionManager">
        <property name="sessionFactory" ref="sessionFactory" />
    </bean>

    <!-- 开启注解事务 -->

```

```

<tx:annotation-driven transaction-manager="transactionManager"/>

<!-- 配置在没有开启类扫描的情况下使用注解 -->
<context:annotation-config />
</beans>

```

- 在相关类使用注解完成属性的注入
- Action中的注入

```

package com.admiral.crm.web.action;

import javax.annotation.Resource;

import com.admiral.crm.domain.SaleVisit;
import com.admiral.crm.service.SaleVisitService;
import com.opensymphony.xwork2.ActionSupport;
import com.opensymphony.xwork2.ModelDriven;

public class SaleVisitAction extends ActionSupport implements
ModelDriven<SaleVisit> {

    //模型驱动使用的对象
    private SaleVisit salevisit = new SaleVisit();

    @Override
    public SaleVisit getModel() {
        return salevisit;
    }

    @Resource(name = "salevisitService")
    private SaleVisitService salevisitService;
}

```

- Service 中的注入

```

package com.admiral.crm.service.impl;

import javax.annotation.Resource;

import com.admiral.crm.dao.SaleVisitDao;
import com.admiral.crm.service.SaleVisitService;

public class SaleVisitServiceImpl implements SaleVisitService {

    @Resource(name = "salevisitDao")
    private SaleVisitDao salevisitDao;
}

```


1.1.3 客户拜访记录的列表显示

1.1.3.1 修改menu.jsp的提交路径

```
<TD class=menuSmall><A class=style2 href="${pageContext.request.contextPath}/saleVisit_findAll.action"
target=main>-客户拜访列表</A></TD>
```

1.1.3.2 编写Action的findAll的方法

```
package com.admiral.crm.web.action;

import java.util.List;

import javax.annotation.Resource;

import org.hibernate.criterion.DetachedCriteria;

import com.admiral.crm.domain.PageBean;
import com.admiral.crm.domain.SaleVisit;
import com.admiral.crm.service.SaleVisitService;
import com.opensymphony.xwork2.ActionContext;
import com.opensymphony.xwork2.ActionSupport;
import com.opensymphony.xwork2.ModelDriven;

public class SaleVisitAction extends ActionSupport implements
ModelDriven<SaleVisit> {

    //模型驱动使用的对象
    private SaleVisit salevisit = new SaleVisit();

    @Override
    public SaleVisit getModel() {
        return salevisit;
    }

    @Resource(name = "saleVisitService")
    private SaleVisitService saleVisitService;

    /**
     * Description: 接收分页的参数
     */
    private Integer currPage = 1;
    private Integer pageSize = 3;
    public void setCurrPage(Integer currPage) {
        if(currPage==null) {
            currPage = 1;
        }
        this.currPage = currPage;
    }
    public void setPageSize(Integer pageSize) {
```

```

        if(pageSize == null) {
            pageSize = 3;
        }
        this.pageSize = pageSize;
    }

    /**
     *
     * Title: findAll
     * Description: 带条件的分页查询的方法
     * @return
     */
    private String findAll() {
        DetachedCriteria detachedCriteria =
        DetachedCriteria.forClass(SaleVisit.class);
        //设置条件

        //调用业务层
        PageBean<SaleVisit> pageBean =
        saleVisitService.findByPage(detachedCriteria, currPage, pageSize);
        //将集合压入值栈
        ActionContext.getContext().getValueStack().push(pageBean);
        return "findAll";
    }
}

```

1.1.3.3 编写Service

```

package com.admiral.crm.service.impl;

import java.util.List;

import javax.annotation.Resource;

import org.hibernate.criterion.DetachedCriteria;

import com.admiral.crm.dao.SaleVisitDao;
import com.admiral.crm.domain.PageBean;
import com.admiral.crm.domain.SaleVisit;
import com.admiral.crm.service.SaleVisitService;

public class SaleVisitServiceImpl implements SaleVisitService {

    @Resource(name = "saleVisitDao")
    private SaleVisitDao saleVisitDao;

    @Override
    public PageBean findByPage(DetachedCriteria detachedCriteria, Integer
currPage, Integer pageSize) {

```

```

//准备 PageBean
PageBean<SaleVisit> pageBean = new PageBean<SaleVisit>();
//设置当前页
pageBean.setCurrPage(currPage);
//设置每页的记录数
pageBean.setPageSize(pageSize);
//设置总记录数
Integer totalCount = saleVisitDao.findCount(detachedCriteria);
pageBean.setTotalCount(totalCount);
//设置总页数
Double num = Math.ceil(totalCount.doubleValue()/pageSize);
pageBean.setTotalPage(num.intValue());

//设置每页要显示的数据的集合
Integer begin = (currPage-1)*pageSize;
List<SaleVisit> list = saleVisitDao.findByPage(detachedCriteria, begin,
pageSize);
pageBean.setList(list);
return pageBean;
}
}

```

1.1.3.4 在页面显示数据

- 配置跳转

```

<!-- 配置客户拜访记录模块 -->
<action name="saleVisit_" class="saleVisitAction" method="{1}">
    <result name="findAll">/jsp/salevisit/list.jsp</result>
</action>

```

- 准备页面

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib uri="/struts-tags" prefix="s"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<TITLE>客户拜访记录列表</TITLE>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<LINK href="${pageContext.request.contextPath }/css/Style.css" type="text/css"
rel="stylesheet">
<LINK href="${pageContext.request.contextPath }/css/Manage.css" type="text/css"
rel="stylesheet">
<script type="text/javascript" src="${pageContext.request.contextPath
}/js/jquery-1.4.4.min.js"></script>
<SCRIPT language="javascript">
    function to_page(page){

```

```

        if(page){
            $("#page").val(page);
        }
        document.customerForm.submit();
    }
</SCRIPT>

<META content="MSHTML 6.00.2900.3492" name=GENERATOR>
</HEAD>
<BODY>
    <s:debug></s:debug>
    <FORM id="customerForm" name="customerForm"
        action="${pageContext.request.contextPath }/saleVisit_findAll.action"
        method=post>

        <TABLE cellSpacing=0 cellPadding=0 width="98%" border=0>
            <TBODY>
                <TR>
                    <TD width=15><IMG src="${pageContext.request.contextPath
}/images/new_019.jpg"
                        border=0></TD>
                    <TD width="100%"
background="${pageContext.request.contextPath }/images/new_020.jpg"
                        height=20></TD>
                    <TD width=15><IMG src="${pageContext.request.contextPath
}/images/new_021.jpg"
                        border=0></TD>
                </TR>
            </TBODY>
        </TABLE>
        <TABLE cellSpacing=0 cellPadding=0 width="98%" border=0>
            <TBODY>
                <TR>
                    <TD width=15 background="${pageContext.request.contextPath
}/images/new_022.jpg"><IMG
                        src="${pageContext.request.contextPath
}/images/new_022.jpg" border=0></TD>
                    <TD valign=top width="100%" bgColor=#ffffff>
                        <TABLE cellSpacing=0 cellPadding=5 width="100%"
border=0>
                            <TR>
                                <TD class=manageHead>当前位置：客户拜访记录管理 &gt;
客户拜访记录列表</TD>
                            </TR>
                        </TABLE>
                    </TD>
                </TR>
            </TBODY>
        </TABLE>
    </FORM>

```

```

                                <TD>联系人名称: </TD>
                                <TD>
                                    <s:textfield
theme="simple" class="textbox" id="schannel2" cssStyle="WIDTH: 80px"
maxLength="50" name="lkm_name"/>
                                </TD>
                                <TD>联系人性别: </TD>
                                <TD>
                                    <s:select theme="simple"
list="#{'1':'男','2':'女'}" name="lkm_gender" headerKey="" headerValue="-请选择-"
/>
                                </TD>

                                <TD><INPUT class=button

id=sButton2 type=submit

name=sButton2></TD>
                                value=" 筛选 "

                                </TR>
                                </TBODY>
                                </TABLE>
                                </TD>
                                </TR>

                                <TR>
                                <TD>
                                    <TABLE id=grid
                                        style="BORDER-TOP-WIDTH: 0px; FONT-
WEIGHT: normal; BORDER-LEFT-WIDTH: 0px; BORDER-LEFT-COLOR: #cccccc; BORDER-
BOTTOM-WIDTH: 0px; BORDER-BOTTOM-COLOR: #cccccc; WIDTH: 100%; BORDER-TOP-COLOR:
#cccccc; FONT-STYLE: normal; BACKGROUND-COLOR: #cccccc; BORDER-RIGHT-WIDTH: 0px;
TEXT-DECORATION: none; BORDER-RIGHT-COLOR: #cccccc"
                                        cellSpacing=1 cellPadding=2
rules=all border=0>
                                <TBODY>
                                    <TR
                                        style="FONT-WEIGHT: bold;
FONT-STYLE: normal; BACKGROUND-COLOR: #eeeeee; TEXT-DECORATION: none">
                                        <TD>业务员名称</TD>
                                        <TD>客户名称</TD>
                                        <TD>拜访时间</TD>
                                        <TD>拜访地点</TD>
                                        <TD>拜访详情</TD>
                                        <TD>下次拜访时间</TD>
                                        <TD>操作</TD>
                                    </TR>
                                    <s:iterator value="list">
                                    <TR
                                        style="FONT-WEIGHT: normal;
FONT-STYLE: normal; BACKGROUND-COLOR: white; TEXT-DECORATION: none">
                                        <TD><s:property

value="user.user_name"/></TD>

                                        <TD>

value="customer.cust_name"/>

                                        </TD>
                                        <TD>

name="visit_time" format="yyyy-MM-dd"/>

```

```

        </TD>
        <TD><s:property
value="visit_addr"/></TD>

        <TD><s:property
value="visit_detail"/></TD>


        <TD>
            <s:date
name="visit_time" format="yyyy-MM-dd"/>

        </TD>
        <TD>
        <a
href="{pageContext.request.contextPath }/">修改</a>
                &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
                <a
href="{pageContext.request.contextPath }/">删除</a>
        </TD>
    </TR>

    </s:iterator>

    </TBODY>
</TABLE>

</TD>
</TR>

<TR>
    <TD><SPAN id=pageLink>
        <DIV style="LINE-HEIGHT: 20px; HEIGHT: 20px; TEXT-ALIGN: right">
            共[<B><s:property value="totalCount"/></B>]条记录,[<B><s:property value="totalPage"/></B>]页,每页显示

            <select name="pageSize" onchange="to_page()">
                <option value="3" <s;if test="pageSize == 3">selected</s:if>>3</option>
                <option value="5" <s;if test="pageSize == 5">selected</s:if>>5</option>
                <option value="10" <s;if test="pageSize == 10">selected</s:if>>10</option>
            </select>
            条
            <s:if test="currPage != 1">
                [<A href="javascript:to_page(<s:property value='1' />)">首页</A>]
                [<A href="javascript:to_page(<s:property value='currPage-1' />)">前一页</A>]
            </s:if>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
            <B>

            <s:iterator var="i" begin="1" end="totalPage">

                <s:if test="#i == currPage">

                    <s:property value="#i"/>

```

```

                                </s:if>
                                <s:else>
                                    <a
href="javascript:to_page(<s:property value="#i" />)"><s:property value="#i"/>
</a>

                                </s:else>
                                </s:iterator>

                                </B>&nbsp;&nbsp; 
                                <s:if test="currPage !=
totalPage">

                                    [<A
href="javascript:to_page(<s:property value="currPage+1" />)">后一页</A>]
                                    [<A
href="javascript:to_page(<s:property value="totalPage" />)">尾页</A>]
                                </s:if>
                                到
                                <input type="text" size="3"
id="page" name="currPage" />
                                页

                                <input type="button" value="Go"
onclick="to_page()" />

                                </DIV>
                                </SPAN></TD>
                            </TR>
                        </TBODY>
                    </TABLE>
                </TD>
                <TD width=15 background="${pageContext.request.contextPath
}/images/new_023.jpg"><IMG
                    src="${pageContext.request.contextPath
}/images/new_023.jpg" border=0></TD>
                </TR>
            </TBODY>
        </TABLE>
        <TABLE cellSpacing=0 cellPadding=0 width="98%" border=0>
            <TBODY>
                <TR>
                    <TD width=15><IMG src="${pageContext.request.contextPath
}/images/new_024.jpg"
                        border=0></TD>
                    <TD align=middle width="100%"
                        background="${pageContext.request.contextPath
}/images/new_025.jpg" height=15></TD>
                    <TD width=15><IMG src="${pageContext.request.contextPath
}/images/new_026.jpg"
                        border=0></TD>
                </TR>
            </TBODY>
        </TABLE>
    </FORM>
</BODY>
</HTML>

```

1.1.4 客户拜访记录的保存

1.1.4.1 修改menu.jsp的链接地址

```
<TD class=menuSmall><A class=style2 href="{pageContext.request.contextPath}/saleVisit_saveUI.action" target=main>—新增客户拜访</A></TD>
```

1.1.4.2 编写Action中的saveUI方法

```
/**
 *
 * Title: saveUI
 * Description: 客户拜访记录管理模块:跳转到添加页面的方法
 * @return
 */
public String saveUI() {

    return "saveUI";
}
```

1.1.4.3 跳转到添加页面

- 配置跳转

```
<!-- 配置客户拜访记录模块 -->
<action name="salevisit_*" class="saleVisitAction" method="{1}">
    <result name="findAll">/jsp/salevisit/list.jsp</result>
    <result name="saveUI">/jsp/salevisit/add.jsp</result>
</action>
```

- 准备添加页面

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib uri="/struts-tags" prefix="s"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<TITLE>添加客户拜访记录</TITLE>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```



```

<LINK href="${pageContext.request.contextPath }/css/Style.css" type=text/css
rel=stylesheet>
<LINK href="${pageContext.request.contextPath }/css/Manage.css" type=text/css
rel=stylesheet>

<META content="MSHTML 6.00.2900.3492" name=GENERATOR>
</HEAD>
<BODY>
    <s:form id="form1" name="form1" action="linkMan_save" method="post"
namespace="/" theme="simple">

        <TABLE cellSpacing=0 cellPadding=0 width="98%" border=0>
            <TBODY>
                <TR>
                    <TD width=15><IMG src="${pageContext.request.contextPath
}/images/new_019.jpg"
                        border=0></TD>
                    <TD width="100%"
background="${pageContext.request.contextPath }/images/new_020.jpg"
                        height=20></TD>
                    <TD width=15><IMG src="${pageContext.request.contextPath
}/images/new_021.jpg"
                        border=0></TD>
                </TR>
            </TBODY>
        </TABLE>
        <TABLE cellSpacing=0 cellPadding=0 width="98%" border=0>
            <TBODY>
                <TR>
                    <TD width=15 background=${pageContext.request.contextPath
}/images/new_022.jpg><IMG
                        src="${pageContext.request.contextPath
}/images/new_022.jpg" border=0></TD>
                    <TD valign=top width="100%" bgColor=#ffffff>
                        <TABLE cellSpacing=0 cellPadding=5 width="100%"
border=0>
                            <TR>
                                <TD class=manageHead>当前位置： 客户拜访记录管理 &gt;
添加客户拜访记录</TD>
                            </TR>
                            <TR>
                                <TD height=2></TD>
                            </TR>
                        </TABLE>
                        <TABLE cellSpacing=0 cellPadding=5 border=0>
                            <tr>
                                <td>拜访客户： </td>
                                <td>
                                    <select name="customer.cust_id"
id="customer">
                                        <option value="">-请选择-</option>
                                    </select>
                                </td>
                                <td>业务员名称： </td>
                                <td>
                                    <select name="user.user_id" id="user">
                                        <option value="">-请选择-</option>

```

```

        </select>
    </td>
</tr>
<TR>
    <td>拜访时间: </td>
    <td>
        <s:textfield name="visit_name"
class="textbox" id="sChannel2" cssStyle="WIDTH: 180px" maxLength="50"/>
    </td>
    <td>拜访地点 : </td>
    <td>
        <s:textfield class="textbox" id="sChannel2"
cssStyle="WIDTH: 180px" maxLength="50" name="visit_addr"/>
    </td>
</TR>
<TR>
    <td>拜访详情 : </td>
    <td>
        <s:textfield class="textbox" id="sChannel2"
cssStyle="WIDTH: 180px" maxLength="50" name="visit_detail"/>
    </td>
    <td>下次拜访时间 : </td>
    <td>
        <s:textfield class="textbox" id="sChannel2"
cssStyle="WIDTH: 180px" maxLength="50" name="visit_nexttime"/>
    </td>
</TR>
<tr>
    <td rowspan=2>
        <INPUT class=button id=sButton2 type=submit
value="保存 "
name=sButton2>
    </td>
</tr>
</TABLE>

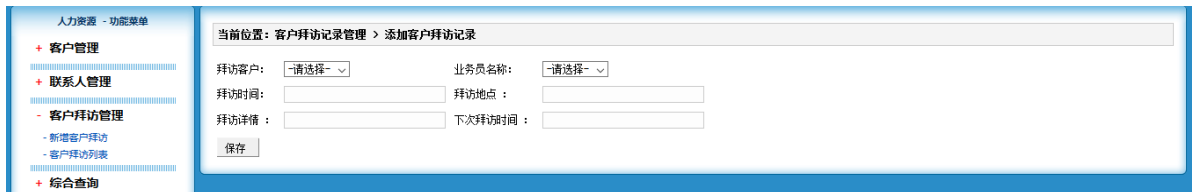
</TD>
<TD width=15 background="${pageContext.request.contextPath
}/images/new_023.jpg">
    <IMG src="${pageContext.request.contextPath
}/images/new_023.jpg" border=0></TD>
</TR>
</TBODY>
</TABLE>
<TABLE cellspacing=0 cellpadding=0 width="98%" border=0>
    <TBODY>
        <TR>
            <TD width=15><IMG src="${pageContext.request.contextPath
}/images/new_024.jpg"
border=0></TD>
            <TD align=middle width="100%"
background="${pageContext.request.contextPath
}/images/new_025.jpg" height=15></TD>
            <TD width=15><IMG src="${pageContext.request.contextPath
}/images/new_026.jpg"
border=0></TD>

```

```

        </TR>
    </TBODY>
</TABLE>
</s:form>
</BODY>
</HTML>

```



1.1.4.4 异步加载数据

- 异步加载客户信息

```

<script type="text/javascript" src="${pageContext.request.contextPath
}/js/jquery-1.11.3.min.js"></script>
<script type="text/javascript">
    $(function() {
        //异步加载客户信息
        $.post("${pageContext.request.contextPath
}/customer_findAllCustomer.action",{ },function(data){
            $(data).each(function(i,n){
                $("#customer").append("<option value='"+ n.cust_id +"'>" +
n.cust_name + "</option>");
            })
        }, "json");
        //异步加载用户信息
    });
</script>

```

- 在 CustomerAction 中提供 findAllCustomer 方法

```

/**
 *
 * Title: findAllCustomer
 * Description: 客户拜访记录管理模块需要用到的异步加载所有客户信息
 * @return
 * @throws IOException
 */
public String findAllCustomer() throws IOException {
    List<Customer> list = customerService.findAll();

    JsonConfig jsonConfig = new JsonConfig();
    jsonConfig.setExcludes(new String[]
{"linkMans","baseDictSource","baseDictLevel","baseDictIndustry"});
    JSONArray jsonArray = JSONArray.fromObject(list,jsonConfig);

```

```

        //将JSON写入到页面

        ServletActionContext.getResponse().setContentType("text/html;charset=UTF-8");

        ServletActionContext.getResponse().getWriter().println(jsonArray.toString());

        return NONE;
    }

```

- 异步加载用户信息

```

<script type="text/javascript" src="${pageContext.request.contextPath
}/js/jquery-1.11.3.min.js"></script>
<script type="text/javascript">
    $(function() {
        //异步加载客户信息
        $.post("${pageContext.request.contextPath
}/customer_findAllCustomer.action",{},{function(data){
            $(data).each(function(i,n){
                $("#customer").append("<option value='"+ n.cust_id +"'>" +
n.cust_name + "</option>");
            })
        },"json");
        //异步加载用户信息
        $.post("${pageContext.request.contextPath }/user_findAllUser.action",
{},{function(data){
            $(data).each(function(i,n){
                $("#user").append("<option value='"+ n.user_id +"'>" +
n.user_name + "</option>");
            })
        },"json");
    });
</script>

```

- 在 UserAction 中提供 findAllUser 的方法

```

/**
 *
 * Title: findAllUser
 * Description: 客户拜访记录管理模块需要用到的异步加载所有客户信息
 * @return
 * @throws IOException
 */
public String findAllUser() throws IOException {
    List<User> list = userService.findAll();

    JSONArray jsonArray = JSONArray.fromObject(list);

    ServletActionContext.getResponse().setContentType("text/html;charset=UTF-8");

    ServletActionContext.getResponse().getWriter().println(jsonArray.toString());

    return NONE;
}

```

UserServiceImpl.java

```
@Override
public List<User> findAll() {
    return userDao.findAll();
}
```

1.1.4.5 编写Action的save方法

```
package com.admiral.crm.web.action;

import javax.annotation.Resource;

import org.hibernate.criterion.DetachedCriteria;

import com.admiral.crm.domain.PageBean;
import com.admiral.crm.domain.SaleVisit;
import com.admiral.crm.service.SaleVisitService;
import com.opensymphony.xwork2.ActionContext;
import com.opensymphony.xwork2.ActionSupport;
import com.opensymphony.xwork2.ModelDriven;

public class SaleVisitAction extends ActionSupport implements
ModelDriven<SaleVisit> {

    //模型驱动使用的对象
    private SaleVisit salevisit = new SaleVisit();

    @Override
    public SaleVisit getModel() {
        return salevisit;
    }

    @Resource(name = "salevisitService")
    private SaleVisitService salevisitService;

    /**
     * Description: 接收分页的参数
     */
    private Integer currPage = 1;
    private Integer pageSize = 3;
    public void setCurrPage(Integer currPage) {
        if(currPage==null) {
            currPage = 1;
        }
        this.currPage = currPage;
    }
}
```

```

public void setPageSize(Integer pageSize) {
    if(pageSize == null) {
        pageSize = 3;
    }
    this.pageSize = pageSize;
}

/**
 *
 * Title: findAll
 * Description: 带条件的分页查询的方法
 * @return
 */
public String findAll() {
    DetachedCriteria detachedCriteria =
DetachedCriteria.forClass(Salevisit.class);
    //设置条件

    //调用业务层
    PageBean<Salevisit> pageBean =
salevisitService.findByPage(detachedCriteria, currPage, pageSize);
    //将集合压入值栈
    ActionContext.getContext().getValueStack().push(pageBean);
    return "findAll";
}

/**
 *
 * Title: saveUI
 * Description: 客户拜访记录管理模块:跳转到添加页面的方法
 * @return
 */
public String saveUI() {

    return "saveUI";
}

/**
 *
 * Title: save
 * Description: 客户拜访记录管理模块:添加拜访记录的方法
 * @return
 */
public String save() {
    salevisitService.save(salevisit);
    return "saveSuccess";
}
}

```

1.1.4.6 编写Service

```
package com.admiral.crm.service.impl;

import java.util.List;

import javax.annotation.Resource;

import org.hibernate.criterion.DetachedCriteria;
import org.springframework.transaction.annotation.Transactional;

import com.admiral.crm.dao.SaleVisitDao;
import com.admiral.crm.domain.PageBean;
import com.admiral.crm.domain.SaleVisit;
import com.admiral.crm.service.SaleVisitService;

@Transactional
public class SaleVisitServiceImpl implements SaleVisitService {

    @Resource(name = "saleVisitDao")
    private SaleVisitDao saleVisitDao;

    @Override
    public PageBean findByPage(DetachedCriteria detachedCriteria, Integer
currPage, Integer pageSize) {

        //准备 PageBean
        PageBean<SaleVisit> pageBean = new PageBean<SaleVisit>();
        //设置当前页
        pageBean.setCurrPage(currPage);
        //设置每页的记录数
        pageBean.setPageSize(pageSize);
        //设置总记录数
        Integer totalCount = saleVisitDao.findCount(detachedCriteria);
        pageBean.setTotalCount(totalCount);
        //设置总页数
        Double num = Math.ceil(totalCount.doubleValue()/pageSize);
        pageBean.setTotalPage(num.intValue());

        //设置每页要显示的数据的集合
        Integer begin = (currPage-1)*pageSize;
        List<SaleVisit> list = saleVisitDao.findByPage(detachedCriteria, begin,
pageSize);
        pageBean.setList(list);
        return pageBean;
    }

    @Override
    public void save(SaleVisit salevisit) {
        saleVisitDao.save(salevisit);
    }

}
```

- 配置跳转

```
<!-- 配置客户拜访记录模块 -->
<action name="salevisit_*" class="salevisitAction" method="{1}">
    <result name="findAll">/jsp/salevisit/list.jsp</result>
    <result name="saveUI">/jsp/salevisit/add.jsp</result>
    <result name="saveSuccess"
type="redirectAction">salevisit_findAll.action</result>
</action>
```

1.1.5 客户拜访记录的条件查询

1.1.5.1 修改了Action的findAll的方法

1.3.2.3 第七步：Action调用业务层

1.3.2.4 第八步：在Service中调用DAO

1.3.2.5 第九步：创建实体和映射（映射使用的是注解）

1.3.2.6 第十步：在Spring中整合Hibernate

1.3.2.7 第十一步：在DAO中使用模板

1.3.2.8 第十二步：配置事务管理