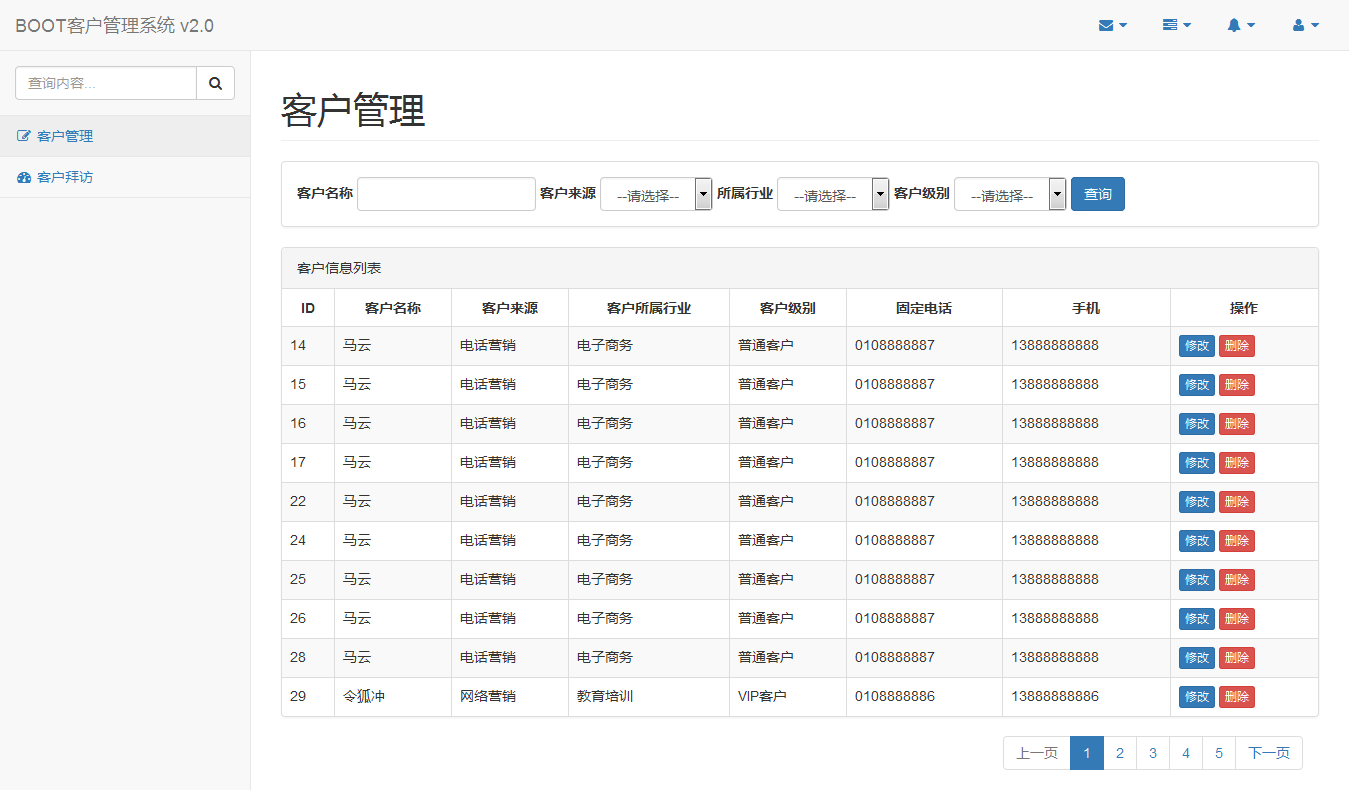
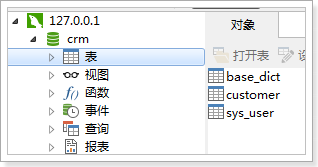
# CRM Web page:



# Database:

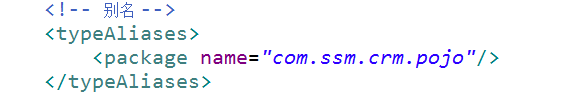
SQL table please refer to attached sql script in public email



# Project:

工程使用Springmvc、spring、mybatis框架整合完成。

### SqlMapConfig.xml



### applicationContext-dao.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:context=*"http://www.springframework.org/schema/context"* xmlns:p=*"http://www.springframework.org/schema/p"*

xmlns:aop=*"http://www.springframework.org/schema/aop"* xmlns:tx=*"http://www.springframework.org/schema/tx"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.0.xsd*

*http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.0.xsd*

*http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-4.0.xsd http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-4.0.xsd*

*http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util-4.0.xsd"*>

<!-- 配置 读取properties文件 jdbc.properties -->

<context:property-placeholder location=*"classpath:jdbc.properties"* />

<!-- 配置 数据源 -->

<bean id=*"dataSource"* class=*"com.alibaba.druid.pool.DruidDataSource"*>

<property name=*"driverClassName"* value=*"${jdbc.driver}"* />

<property name=*"url"* value=*"${jdbc.url}"* />

<property name=*"username"* value=*"${jdbc.username}"* />

<property name=*"password"* value=*"${jdbc.password}"* />

</bean>

<!-- 配置SqlSessionFactory -->

<bean class=*"org.mybatis.spring.SqlSessionFactoryBean"*>

<!-- 设置MyBatis核心配置文件 -->

<property name=*"configLocation"* value=*"classpath:mybatis/SqlMapConfig.xml"* />

<!-- 设置数据源 -->

<property name=*"dataSource"* ref=*"dataSource"* />

</bean>

<!-- 配置Mapper扫描 -->

<bean class=*"org.mybatis.spring.mapper.MapperScannerConfigurer"*>

<!-- 设置Mapper扫描包 -->

<property name=*"basePackage"* value=*"cn.itcast.crm.mapper"* />

</bean>

</beans>

### jdbc.properties

jdbc.driver=com.mysql.jdbc.Driver

jdbc.url=jdbc:mysql://localhost:3306/crm?characterEncoding=utf-8

jdbc.username=root

jdbc.password=root

### log4j.properties

# Global logging configuration

log4j.rootLogger=DEBUG, stdout

# Console output...

log4j.appender.stdout=org.apache.log4j.ConsoleAppender

log4j.appender.stdout.layout=org.apache.log4j.PatternLayout

log4j.appender.stdout.layout.ConversionPattern=%5p [%t] - %m%n

### applicationContext-service.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:context=*"http://www.springframework.org/schema/context"* xmlns:p=*"http://www.springframework.org/schema/p"*

xmlns:aop=*"http://www.springframework.org/schema/aop"* xmlns:tx=*"http://www.springframework.org/schema/tx"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.0.xsd*

*http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.0.xsd*

*http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-4.0.xsd http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-4.0.xsd*

*http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util-4.0.xsd"*>

<!-- 配置Service扫描 -->

<context:component-scan base-package=*"cn.itcast.crm.service"* />

</beans>

### 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:p=*"http://www.springframework.org/schema/p"*

xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:mvc=*"http://www.springframework.org/schema/mvc"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.0.xsd*

*http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-4.0.xsd*

*http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.0.xsd"*>

<!-- 配置Controller扫描 -->

<context:component-scan base-package=*"cn.itcast.crm.controller"* />

<!-- 配置注解驱动 -->

<mvc:annotation-driven />

<!-- 配置视图解析器 -->

<bean class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>

<!-- 前缀 -->

<property name=*"prefix"* value=*"/WEB-INF/jsp/"* />

<!-- 后缀 -->

<property name=*"suffix"* value=*".jsp"* />

</bean>

</beans>

### 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\_2\_5.xsd"*

id=*"WebApp\_ID"* version=*"2.5"*>

<display-name>boot-crm</display-name>

<welcome-file-list>

<welcome-file>index.jsp</welcome-file>

</welcome-file-list>

<!-- 配置spring -->

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>classpath:spring/applicationContext-\*.xml</param-value>

</context-param>

<!-- 配置监听器加载spring -->

<listener>

<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

</listener>

<!-- 配置过滤器，解决post的乱码问题 -->

<filter>

<filter-name>encoding</filter-name> <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>

</filter>

<filter-mapping>

<filter-name>encoding</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<!-- 配置SpringMVC -->

<servlet>

<servlet-name>boot-crm</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<init-param>

<param-name>contextConfigLocation</param-name>

<param-value>classpath:spring/springmvc.xml</param-value>

</init-param>

<!-- 配置springmvc什么时候启动，参数必须为整数 -->

<!-- 如果为0或者大于0，则springMVC随着容器启动而启动 -->

<!-- 如果小于0，则在第一次请求进来的时候启动 -->

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>boot-crm</servlet-name>

<!-- 所有的请求都进入springMVC -->

<url-pattern>/</url-pattern>

</servlet-mapping>

</web-app>

# JSP Web page:

Static jsp web page is available in public mail attachment.

# Query condition:

## Requirement:



Web jsp:

<form class=*"form-inline"* action=*"*${pageContext.request.contextPath }*/customer/list.action"* method=*"get"*>

<div class=*"form-group"*>

<label for=*"customerName"*>客户名称</label>

<input type=*"text"* class=*"form-control"* id=*"customerName"* value=*"*${custName }*"* name=*"custName"*>

</div>

<div class=*"form-group"*>

<label for=*"customerFrom"*>客户来源</label>

<select class=*"form-control"* id=*"customerFrom"* placeholder=*"客户来源"* name=*"custSource"*>

<option value=*""*>--请选择--</option>

<c:forEach items=*"*${fromType}*"* var=*"item"*>

<option value=*"*${item.dict\_id}*"*<c:if test="${item.dict\_id == custSource}"> selected</c:if>>${item.dict\_item\_name }</option>

</c:forEach>

</select>

</div>

<div class=*"form-group"*>

<label for=*"custIndustry"*>所属行业</label>

<select class=*"form-control"* id=*"custIndustry"* name=*"custIndustry"*>

<option value=*""*>--请选择--</option>

<c:forEach items=*"*${industryType}*"* var=*"item"*>

<option value=*"*${item.dict\_id}*"*<c:if test="${item.dict\_id == custIndustry}"> selected</c:if>>${item.dict\_item\_name }</option>

</c:forEach>

</select>

</div>

<div class=*"form-group"*>

<label for=*"custLevel"*>客户级别</label>

<select class=*"form-control"* id=*"custLevel"* name=*"custLevel"*>

<option value=*""*>--请选择--</option>

<c:forEach items=*"*${levelType}*"* var=*"item"*>

<option value=*"*${item.dict\_id}*"*<c:if test="${item.dict\_id == custLevel}"> selected</c:if>>${item.dict\_item\_name }</option>

</c:forEach>

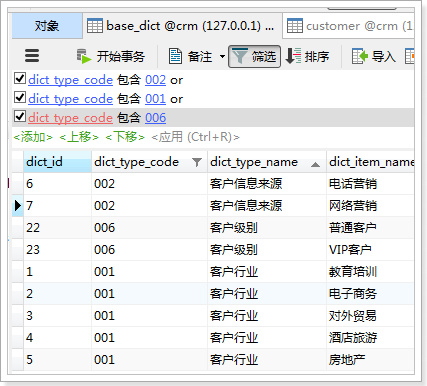
</select>

</div>

<button type=*"submit"* class=*"btn btn-primary"*>查询</button>

</form>

Database query:



Refer sql:

SELECT \* FROM base\_dict WHERE dict\_type\_code = '001'

## Implement DAO

### pojo

**public** **class** BaseDict {

**private** String dict\_id;

**private** String dict\_type\_code;

**private** String dict\_type\_name;

**private** String dict\_item\_name;

**private** String dict\_item\_code;

**private** Integer dict\_sort;

**private** String dict\_enable;

**private** String dict\_memo;

get/set。。。。。。

}

### Mapper

**public** **interface** BaseDictMapper {

/\*\*

\* 根据类别代码查询数据

\*

\* **@param** dictTypeCode

\* **@return**

\*/

List<BaseDict> queryBaseDictByDictTypeCode(String dictTypeCode);

}

### Mapper.xml

<?xml version=*"1.0"* encoding=*"UTF-8"* ?>

<!DOCTYPE mapper

PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"

"http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace=*"cn.itcast.crm.mapper.BaseDictMapper"*>

<!-- 根据类别代码查询数据 -->

<select id=*"queryBaseDictByDictTypeCode"* parameterType=*"String"*

resultType=*"cn.itcast.crm.pojo.BaseDict"*>

SELECT \* FROM base\_dict WHERE dict\_type\_code =

#{dict\_type\_code}

</select>

</mapper>

## Implement Service

### BaseDictService

**public** **interface** BaseDictService {

/\*\*

\* 根据类别代码查询

\*

\* **@param** dictTypeCode

\* **@return**

\*/

List<BaseDict> queryBaseDictByDictTypeCode(String dictTypeCode);

}

### BaseDictServiceImpl

@Service

**public** **class** BaseDictServiceImpl **implements** BaseDictService {

@Autowired

**private** BaseDictMapper baseDictMapper;

@Override

**public** List<BaseDict> queryBaseDictByDictTypeCode(String dictTypeCode) {

List<BaseDict> list = **this**.baseDictMapper.queryBaseDictByDictTypeCode(dictTypeCode);

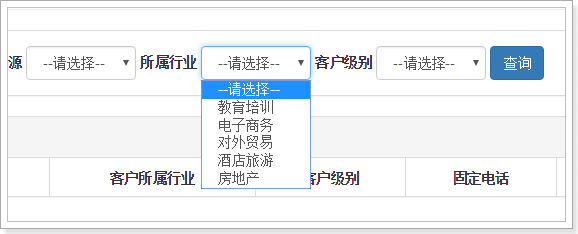
**return** list;

}

}

## Implement Controller

Please implement controller according to class.



## Customer pojo:

**public** **class** Customer {

**private** Long cust\_id;

**private** String cust\_name;

**private** Long cust\_user\_id;

**private** Long cust\_create\_id;

**private** String cust\_source;

**private** String cust\_industry;

**private** String cust\_level;

**private** String cust\_linkman;

**private** String cust\_phone;

**private** String cust\_mobile;

**private** String cust\_zipcode;

**private** String cust\_address;

**private** Date cust\_createtime;

get/set。。。。。。

}

## Implement DAO

### QueryVo

**public** **class** QueryVo {

**private** String custName;

**private** String custSource;

**private** String custIndustry;

**private** String custLevel;

// 当前页码数

**private** Integer page = 1;

// 数据库从哪一条数据开始查

**private** Integer start;

// 每页显示数据条数

**private** Integer rows = 10;

get/set。。。。。。

}

### Mapper

**public** **interface** CustomerMapper {

/\*\*

\* 根据queryVo分页查询数据

\*

\* **@param** queryVo

\* **@return**

\*/

List<Customer> queryCustomerByQueryVo(QueryVo queryVo);

/\*\*

\* 根据queryVo查询数据条数

\*

\* **@param** queryVo

\* **@return**

\*/

**int** queryCountByQueryVo(QueryVo queryVo);

}

### Mapper.xml

<?xml version=*"1.0"* encoding=*"UTF-8"* ?>

<!DOCTYPE mapper

PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"

"http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace=*"cn.itcast.crm.mapper.CustomerMapper"*>

<sql id=*"customerQueryVo"*>

<where>

<if test=*"custName != null and custName != ''"*>

AND a.cust\_name LIKE '%${custName}%'

</if>

<if test=*"custSource != null and custSource != ''"*>

AND a.cust\_source = #{custSource}

</if>

<if test=*"custIndustry != null and custIndustry != ''"*>

AND a.cust\_industry = #{custIndustry}

</if>

<if test=*"custLevel != null and custLevel != ''"*>

AND a.cust\_level = #{custLevel}

</if>

</where>

</sql>

<!-- 根据queryVo分页查询数据 -->

<select id=*"queryCustomerByQueryVo"* parameterType=*"cn.itcast.crm.pojo.QueryVo"*

resultType=*"cn.itcast.crm.pojo.Customer"*>

SELECT

a.cust\_id,

a.cust\_name,

a.cust\_user\_id,

a.cust\_create\_id,

b.dict\_item\_name cust\_source,

c.dict\_item\_name cust\_industry,

d.dict\_item\_name cust\_level,

a.cust\_linkman,

a.cust\_phone,

a.cust\_mobile,

a.cust\_zipcode,

a.cust\_address,

a.cust\_createtime

FROM

customer a

LEFT JOIN base\_dict b ON a.cust\_source = b.dict\_id

LEFT JOIN base\_dict c ON a.cust\_industry = c.dict\_id

LEFT JOIN base\_dict d ON a.cust\_level = d.dict\_id

<include refid=*"customerQueryVo"* />

<if test=*"start != null"*>

LIMIT #{start}, #{rows}

</if>

</select>

<!-- 根据queryVo查询数据条数 -->

<select id=*"queryCountByQueryVo"* parameterType=*"cn.itcast.crm.pojo.QueryVo"*

resultType=*"int"*>

SELECT count(1) FROM customer a

<include refid=*"customerQueryVo"* />

</select>

</mapper>

## service

**public** **interface** CustomerService {

/\*\*

\* 根据条件分页查询客户

\*

\* **@param** queryVo

\* **@return**

\*/

Page<Customer> queryCustomerByQueryVo(QueryVo queryVo);

}

CustomerServiceImpl

@Service

**public** **class** CustomerServiceImpl **implements** CustomerService {

@Autowired

**private** CustomerMapper customerMapper;

@Override

**public** Page<Customer> queryCustomerByQueryVo(QueryVo queryVo) {

// 设置查询条件,从哪一条数据开始查

queryVo.setStart((queryVo.getPage() - 1) \* queryVo.getRows());

// 查询数据结果集

List<Customer> list = **this**.customerMapper.queryCustomerByQueryVo(queryVo);

// 查询到的数据总条数

**int** total = **this**.customerMapper.queryCountByQueryVo(queryVo);

// 封装返回的page对象

Page<Customer> page = **new** Page<>(total, queryVo.getPage(), queryVo.getRows(), list);

**return** page;

}

}

## Update Controller

@RequestMapping("list")

**public** String queryCustomerList(QueryVo queryVo, Model model) {

**try** {

// 解决get请求乱码问题

**if** (StringUtils.*isNotBlank*(queryVo.getCustName())) {

queryVo.setCustName(**new** String(queryVo.getCustName().getBytes("ISO-8859-1"), "UTF-8"));

}

} **catch** (Exception e) {

e.printStackTrace();

}

// 客户来源

List<BaseDict> fromType = **this**.baseDictService.queryBaseDictByDictTypeCode(**this**.CUSTOMER\_FROM\_TYPE);

// 所属行业

List<BaseDict> industryType = **this**.baseDictService.queryBaseDictByDictTypeCode(**this**.CUSTOMER\_INDUSTRY\_TYPE);

// 客户级别

List<BaseDict> levelType = **this**.baseDictService.queryBaseDictByDictTypeCode(**this**.CUSTOMER\_LEVEL\_TYPE);

// 把前端页面需要显示的数据放到模型中

model.addAttribute("fromType", fromType);

model.addAttribute("industryType", industryType);

model.addAttribute("levelType", levelType);

// 分页查询数据

Page<Customer> page = **this**.customerService.queryCustomerByQue ryVo(queryVo);

// 把分页查询的结果放到模型中

model.addAttribute("page", page);

// 数据回显

model.addAttribute("custName", queryVo.getCustName());

model.addAttribute("custSource", queryVo.getCustSource());

model.addAttribute("custIndustry", queryVo.getCustIndustry());

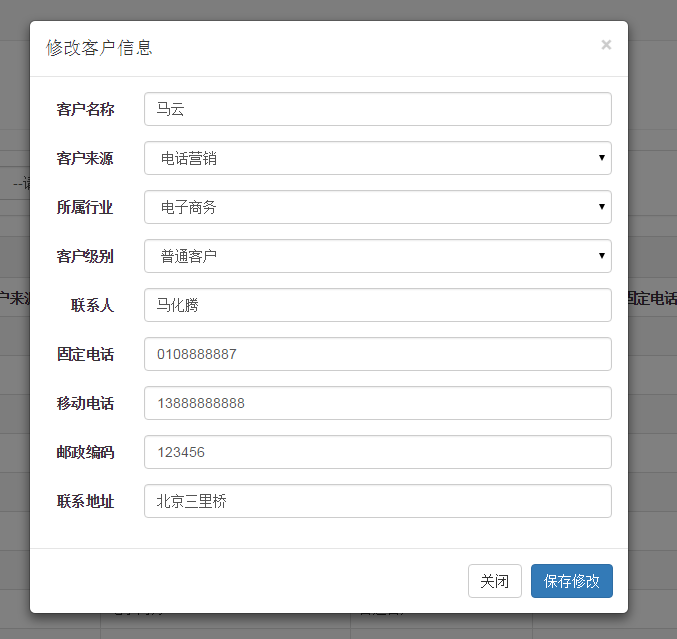
model.addAttribute("custLevel", queryVo.getCustLevel());

**return** "customer";

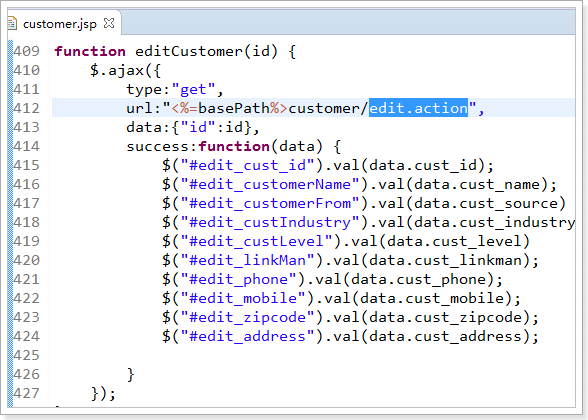
}

## Update Customer:

Requirement:



Ajax for this:



## Echoes implement

### Mapper

/\*\*

\* 根据id查询客户

\*

\* **@param** id

\* **@return**

\*/

Customer queryCustomerById(Long id);

### Mapper.xml

<!-- 根据id查询用户 -->

<select id=*"queryCustomerById"* resultType=*"cn.itcast.crm.pojo.Customer"*>

SELECT \* FROM customer WHERE cust\_id = #{id}

</select>

### Service接口

/\*\*

\* 根据id查询数据

\*

\* **@param** id

\* **@return**

\*/

Customer queryCustomerById(Long id);

### Service接口实现类

@Override

**public** Customer queryCustomerById(Long id) {

Customer customer = **this**.customerMapper.queryCustomerById(id);

**return** customer;

}

### Controller

/\*\*

\* 根据id查询用户,返回json格式数据

\*

\* **@param** id

\* **@return**

\*/

@RequestMapping("edit")

@ResponseBody

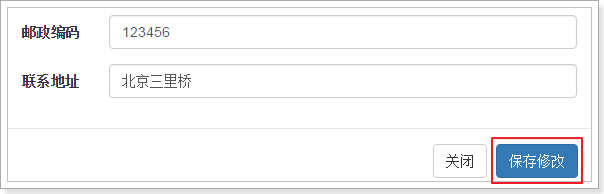
**public** Customer queryCustomerById(Long id) {

Customer customer = **this**.customerService.queryCustomerById(id);

**return** customer;

}

## Edit custom info



**function** updateCustomer() {

$.post("<%=basePath%>customer/update.action",$("#edit\_customer\_form").serialize(),function(data){

alert("客户信息更新成功！");

window.location.reload();

});

}

/\*\*

\* 根据id编辑客户

\*

\* **@param** customer

\*/

**void** updateCustomerById(Customer customer);

### Mapper.xml

CustomerMapper.xml for update customer:

<select id=*"updateCustomerById"* parameterType=*"cn.itcast.crm.pojo.Customer"*>

UPDATE `customer`

SET

<if test=*"cust\_name !=null and cust\_name != ''"*>

`cust\_name` = #{cust\_name},

</if>

<if test=*"cust\_user\_id !=null"*>

`cust\_user\_id` = #{cust\_user\_id},

</if>

<if test=*"cust\_create\_id !=null"*>

`cust\_create\_id` = #{cust\_create\_id},

</if>

<if test=*"cust\_source !=null and cust\_source != ''"*>

`cust\_source` = #{cust\_source},

</if>

<if test=*"cust\_industry !=null and cust\_industry != ''"*>

`cust\_industry` = #{cust\_industry},

</if>

<if test=*"cust\_level !=null and cust\_level != ''"*>

`cust\_level` = #{cust\_level},

</if>

<if test=*"cust\_linkman !=null and cust\_linkman != ''"*>

`cust\_linkman` = #{cust\_linkman},

</if>

<if test=*"cust\_phone !=null and cust\_phone != ''"*>

`cust\_phone` = #{cust\_phone},

</if>

<if test=*"cust\_mobile !=null and cust\_mobile != ''"*>

`cust\_mobile` = #{cust\_mobile},

</if>

<if test=*"cust\_zipcode !=null and cust\_zipcode != ''"*>

`cust\_zipcode` = #{cust\_zipcode},

</if>

<if test=*"cust\_address !=null and cust\_address != ''"*>

`cust\_address` = #{cust\_address},

</if>

`cust\_createtime` = NOW()

WHERE

(`cust\_id` = #{cust\_id});

</select>

### Service

/\*\*

\* 根据id编辑客户数据

\*

\* **@param** customer

\*/

**void** updateCustomerById(Customer customer);

@Override

**public** **void** updateCustomerById(Customer customer) {

**this**.customerMapper.updateCustomerById(customer);

}

# Delete function

Please implement this yourself:

