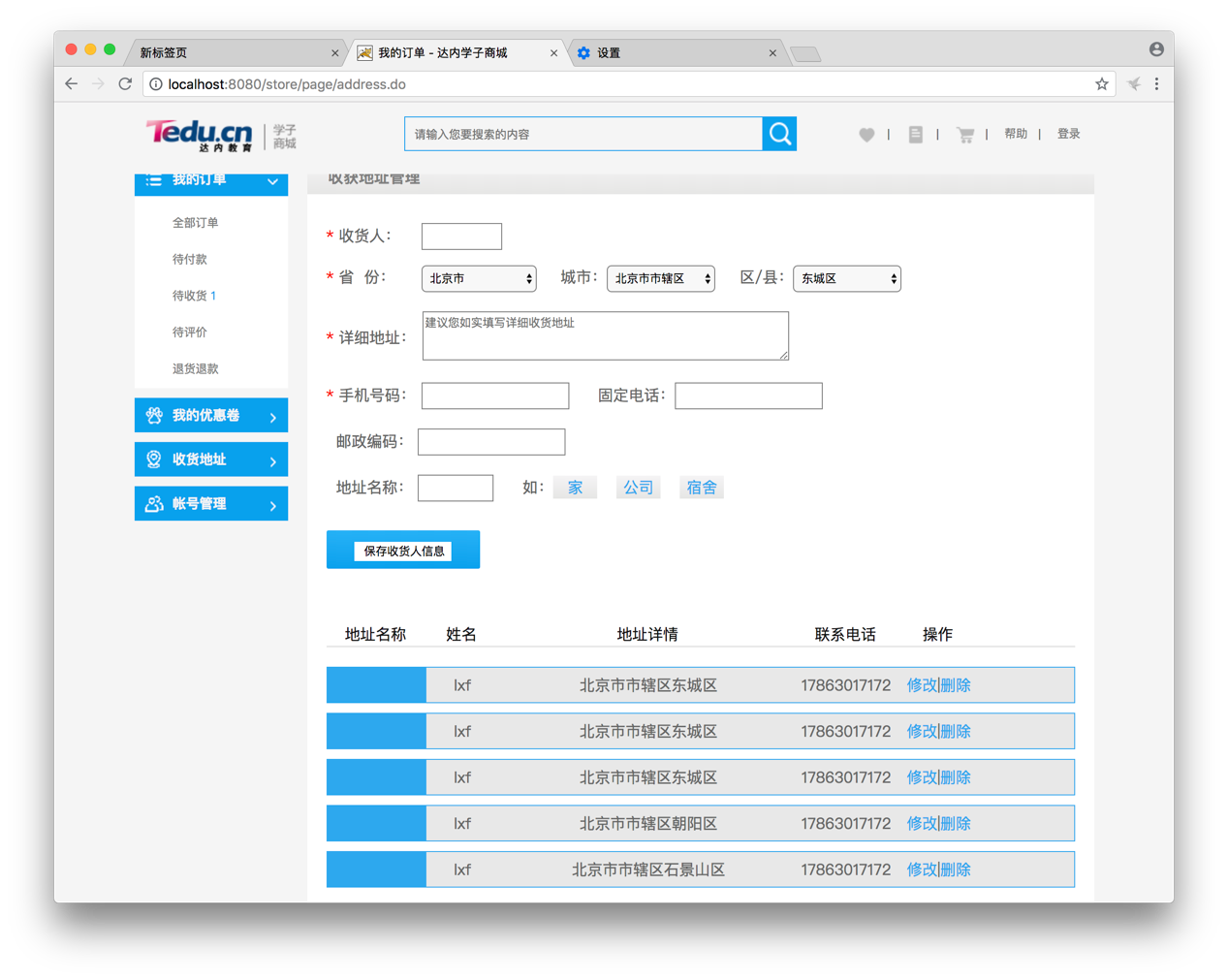
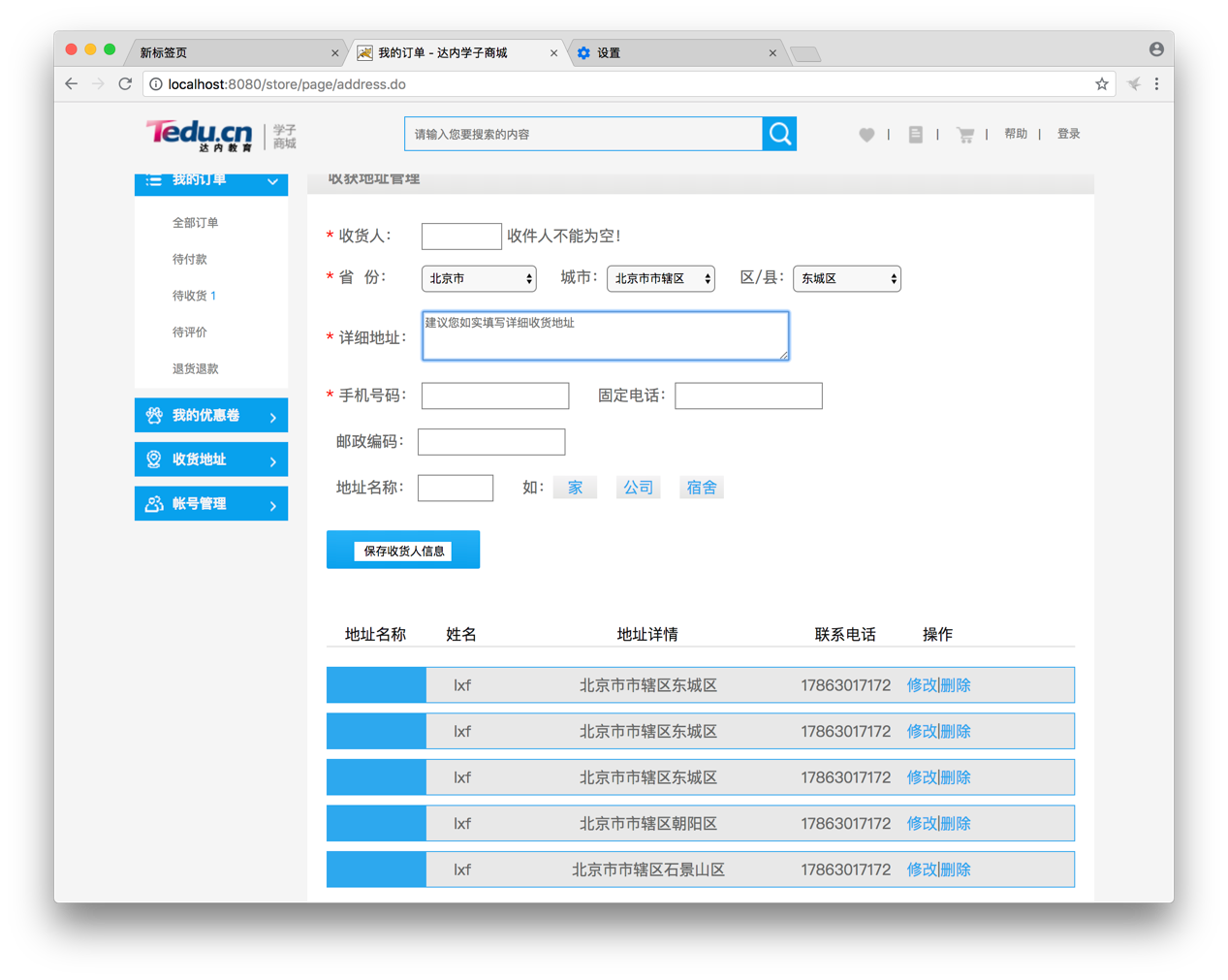
# 功能概述：

用户填写收货地址信息，为文本框注册事件，若未填写必填项目则在页面上提示，在submit后后台也检验是否为空，必填项为空则抛出异常。从session中获取userName，从user表中查找出id作为uid一起填入数据库中，填入后页面进行局部刷新显示填写的的地址信息，页面刚跳转到订单管理页面时，网页根据session中的userName查找数据库中的地址信息显示在页面的下方。

注：数据库连接密码（123456）

效果显示：





# 详细描述：

# 一、实现了保存收货人信息的功能

## 前端对画\*的输入进行提醒，若为空则进行提示（服务器端也有进行判断）

$(**function**(){

$("#form-addressAdmin").submit(commitForm);

**var** userName='${loginUser.name}';

$("#receiverName").blur(checkName);

$("#receiverAddress").blur(checkAddress);

$("#receiverMobile").blur(checkMobile);

showAddresses(userName);

});

**function** checkName(){

**var** name = $("#receiverName").val();

console.log(name);

**if**(name==""||name==**null**){

console.log(name+"hh");

$("#receiverName").next().html("收件人不能为空！").removeClass().addClass("msg-error").show();

}**else**{

$("#receiverName").next().hide();

}

}

## 配置拦截器

直接点击商品页面会弹到登录页面

<!-- 配置 拦截器-->

<mvc:interceptors>

<mvc:interceptor>

<mvc:mapping path=*"/page/\*"*/>

<mvc:exclude-mapping path=*"/user/\*"*/>

<mvc:exclude-mapping path=*"/goods/\*"*/>

<ref bean=*"accessInterceptor"*/>

</mvc:interceptor>

</mvc:interceptors>

## 3.控制器

(1)设置只有登录后才能查看订单管理

@Controller

@RequestMapping("/page")

**public** **class** PageController {

@Resource

**private** IGoodsCategoryService goodsCategoryService;

@Resource

**private** IGoodsService goodsService;

@RequestMapping("/orders.do")

**public** String orders(HttpServletRequest request, ModelMap map) {

**return** "orders";

}

@RequestMapping("/address.do")

**public** String address(HttpServletRequest request, ModelMap map) {

HttpSession session = request.getSession();

User user = (User) session.getAttribute("loginUser");

System.***out***.println(user);

map.put("loginUser", user);

**return** "addressAdmin";

}

}

(2)保存收货人信息

AddressController.java

@Resource

IAddressService addressService;

@RequestMapping("/addressAdmin.do")

@ResponseBody

**public** Result addressAdmin(String userName,String receiverName, String receiverProvince, String receiverCity, String receiverArea,

String receiverAddress,String receiverMobile,String receiverPhone,String receiverZip,String addressName,ModelMap map){

**try**{

System.***out***.println(receiverName+","+receiverProvince+","+receiverCity+receiverArea+","+

receiverAddress+","+receiverMobile+","+receiverPhone+","+receiverZip+addressName);

addressService.addressAdmin(userName,receiverName, receiverProvince, receiverCity, receiverArea,

receiverAddress,receiverMobile,receiverPhone,receiverZip,addressName);

System.***out***.println("addAddress()");

**return** **new** Result("成功");

}**catch**(Exception e){

e.printStackTrace();

map.put("message", e.getMessage());

**return** **new** Result(e);

}

}

## 

## 4.实体类

Address.java

（部分代码）包括无参构造方法，构造方法，toString方法，getter、setter方法，hashcode，equals方法

**private** **static** **final** **long** ***serialVersionUID*** = -6147482980211031221L;

**private** Integer id;

**private** Integer uid;

**private** String recv\_name;

**private** String recv\_province;

**private** String recv\_city;

**private** String recv\_area;

**private** String recv\_district;

**private** String recv\_addr;

**private** String recv\_phone;

**private** String recv\_tel;

**private** String recv\_zip;

**private** String recv\_tag;

**private** Integer is\_default;

**private** String created\_user;

**private** Date created\_time;

**private** String modified\_user;

**private** Date modified\_time;

**public** Address() {

**super**();

}

## 5.业务类

AddressService.java

@Resource

**private** AddressMapper addressMapper;

**public** **void** addressAdmin(String userName,String receiverName, String receiverProvince, String receiverCity, String receiverArea,

String receiverAddress, String receiverMobile,String receiverPhone, String receiverZip, String addressName) {

**if**(receiverName==**null** || receiverName.trim().isEmpty()){

**throw** **new** ServiceException("收货人不能为空1！");

}

**if**(receiverProvince==**null** || receiverProvince.trim().isEmpty()){

**throw** **new** ServiceException("省份不能为空1");

}

**if**(receiverCity==**null** || receiverCity.trim().isEmpty()){

**throw** **new** ServiceException("城市不能为空1");

}

**if**(receiverArea==**null** || receiverArea.trim().isEmpty()){

**throw** **new** ServiceException("区/县不能为空1");

}

**if**(receiverAddress==**null** || receiverAddress.trim().isEmpty()){

**throw** **new** ServiceException("详细地址不能为空1");

}

**if**(receiverMobile==**null** || receiverMobile.trim().isEmpty()){

**throw** **new** ServiceException("手机号码不能为空");

}

**int** length=receiverCity.length();

String str = **null**;

**if** (length >= 3) {

str = receiverCity.substring(length - 3, length);

}

**if** (str.equals("市辖区")) {

receiverCity = "市辖区";

} **else** {

**if** (length >= 2) {

str = receiverCity.substring(length - 2, length);

}

**if** (str.equals("郊县")) {

receiverCity = "县";

}

}

System.***out***.println(receiverCity);

IdList list = addressMapper.findId(receiverProvince,receiverCity,receiverArea);

System.***out***.println(userName);

**int** userId =addressMapper.findUid(userName);

String receiverDistrict=receiverProvince+receiverCity+receiverArea;

**if**(list.getProvince\_code()==**null** || list.getProvince\_code().trim().isEmpty()){

**throw** **new** ServiceException("省份不能为空2");

}

**if**(list.getCity\_code()==**null** || list.getCity\_code().trim().isEmpty()){

**throw** **new** ServiceException("城市不能为空2");

}

**if**(list.getArea\_code()==**null** || list.getArea\_code().trim().isEmpty()){

**throw** **new** ServiceException("区/县不能为空2");

}

Date date = **new** Date();

SimpleDateFormat fmt = **new** SimpleDateFormat("yyyyMMddHHmmss");

String time = fmt.format(date);

Address address = **new** Address(**null**,userId,receiverName,list.getProvince\_code(),list.getCity\_code(),

list.getArea\_code(),receiverDistrict,receiverAddress,receiverMobile,receiverPhone,receiverZip,

addressName,0,userName,date,userName,date);

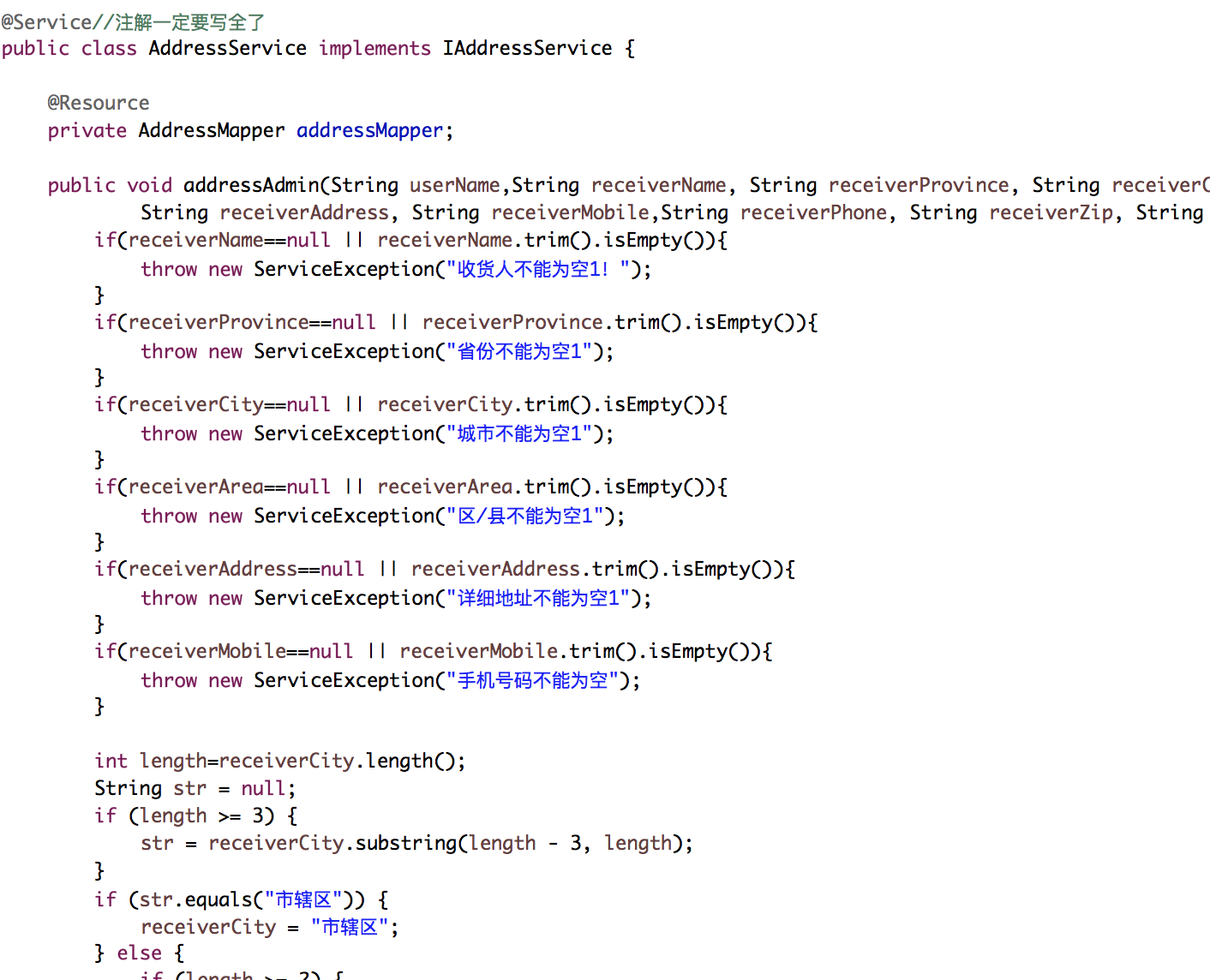
**int** n = addressMapper.addAddress(address);

**if**(n!=1){

**throw** **new** ServiceException("数据保存异常："+n);

}

}



## 6.接口

(1)业务接口

**public** **interface** IAddressService {

**void** addressAdmin(String userName,String receiverName, String receiverProvince, String receiverCity, String receiverArea,

String receiverAddress, String receiverMobile,String receiverPhone, String receiverZip, String addressName);

List<Address> findAddressDetails(String userName);

}

(2)Mapper接口

**public** **interface** AddressMapper {

IdList findId(@Param("receiverProvince")String receiverProvince,

@Param("receiverCity")String receiverCity,

@Param("receiverArea")String receiverArea);

**int** findUid(@Param("userName")String userName);

**int** addAddress(Address address);

List<Address> getAddressDetails(String userName);

}

## 7.Mapper文件

<mapper namespace=*"sdu.edu.store.mapper.AddressMapper"*>

<insert id=*"addAddress"* parameterType=*"sdu.edu.store.entity.Address"*>

insert into t\_address (

id,

uid,

recv\_name,

recv\_province,

recv\_city,

recv\_area,

recv\_district,

recv\_addr,

recv\_phone,

recv\_tel,

recv\_zip,

recv\_tag,

created\_user,

created\_time,

modified\_user,

modified\_time

) values (

#{id},

#{uid},

#{recv\_name},

#{recv\_province},

#{recv\_city},

#{recv\_area},

#{recv\_district},

#{recv\_addr},

#{recv\_phone},

#{recv\_tel},

#{recv\_zip},

#{recv\_tag},

#{created\_user},

#{created\_time},

#{modified\_user},

#{modified\_time}

)

</insert>

<select id=*"findId"* resultType=*"sdu.edu.store.entity.IdList"*>

select

t\_dict\_provinces.province\_code as province\_code,

t\_dict\_cities.city\_code as city\_code,

t\_dict\_areas.area\_code as area\_code

from

t\_dict\_provinces,t\_dict\_cities,t\_dict\_areas

where

t\_dict\_provinces.province\_name=#{receiverProvince} and

t\_dict\_cities.city\_name=#{receiverCity} and

t\_dict\_areas.area\_name=#{receiverArea} and

t\_dict\_provinces.province\_code=t\_dict\_cities.province\_code and

t\_dict\_cities.city\_code=t\_dict\_areas.city\_code

</select>

<select id=*"findUid"* resultType=*"int"*>

select

id as uid

from

user

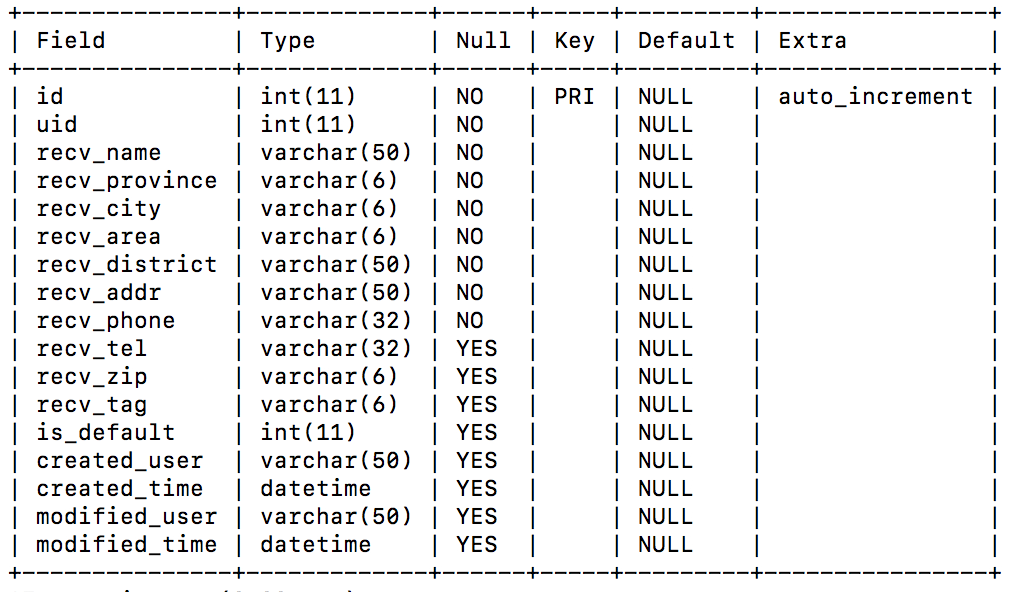
where

name=#{userName}

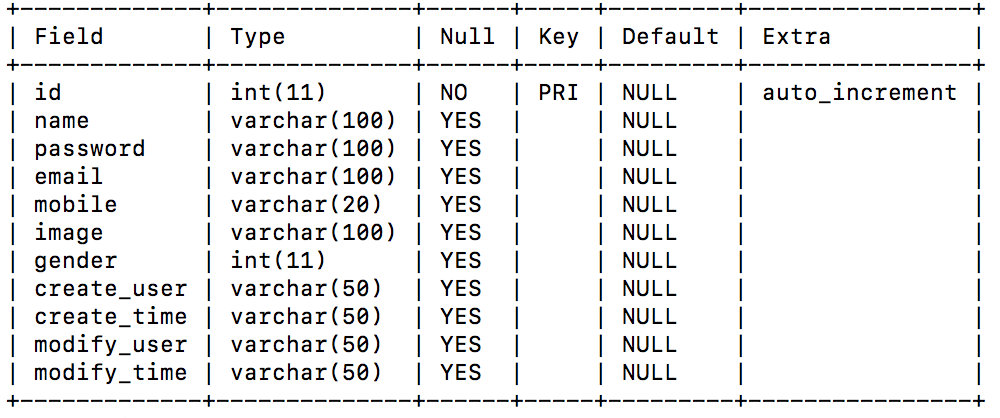
</select>

## 8.数据库表

t\_address表



user表



# 二、实现了ajax异步显示已保存地址信息

## 1.ajax异步更新+JSON传值

**function** showAddresses(userName){

$("#cat161").empty();

**var** title = '<div class="aim\_title">'+

'<span class="dzmc dzmc\_title">地址名称</span>'+

'<span class="dzxm dzxm\_title">姓名</span>'+

'<span class="dzxq dzxq\_title">地址详情</span>'+

'<span class="lxdh lxdh\_title">联系电话</span>'+

' <span class="operation operation\_title">操作</span>'+

' </div>';

$("#cat161").append(title);

**var** url = "../address/addressShow.do";

$.getJSON(url, {"userName":userName}, **function**(result){

**if**(result.state==SUCCESS){

**var** addresses=result.data;

showAddressDetails(addresses);

}**else**{

console.log(result.message);

}

});

}

**function** showAddressDetails(addresses){

**if**(addresses.length>0){

**for**(**var** i=0; i<addresses.length; i++){

**var** address = addresses[i];

**var** template='<div class="aim\_content\_one aim\_active">'+

'<span class="dzmc dzmc\_active">{tag}</span>'+

'<span class="dzxm dzxm\_normal">{name}</span>'+

'<span class="dzxq dzxq\_normal">{place}</span>'+

'<span class="lxdh lxdh\_normal">{phone}</span>'+

'<span class="operation operation\_normal">'+

'<span class="aco\_change">修改</span>|<span class="aco\_delete">删除</span>'+

'</span>'+

'<span class="swmr swmr\_normal"></span>'+

'</div>';

**var** detail = template.replace('{tag}', address.recv\_tag).replace("{name}", address.recv\_name)

.replace('{place}', address.recv\_district).replace("{phone}", address.recv\_phone);

$("#cat161").append(detail);

}

}

}

## 2.控制器

AddressController.java

@RequestMapping("/addressShow.do")

@ResponseBody

**public** Result address(String userName,ModelMap map){

**try**{

List<Address> list = addressService.findAddressDetails(userName);

**for** (Address g : list) {

System.***out***.println(g);

}

**return** **new** Result(list);

}**catch**(Exception e){

e.printStackTrace();

map.put("message", e.getMessage());

**return** **new** Result(e);

}

}

## 3.业务类

AddressService.java

**public** List<Address> findAddressDetails(String userName) {

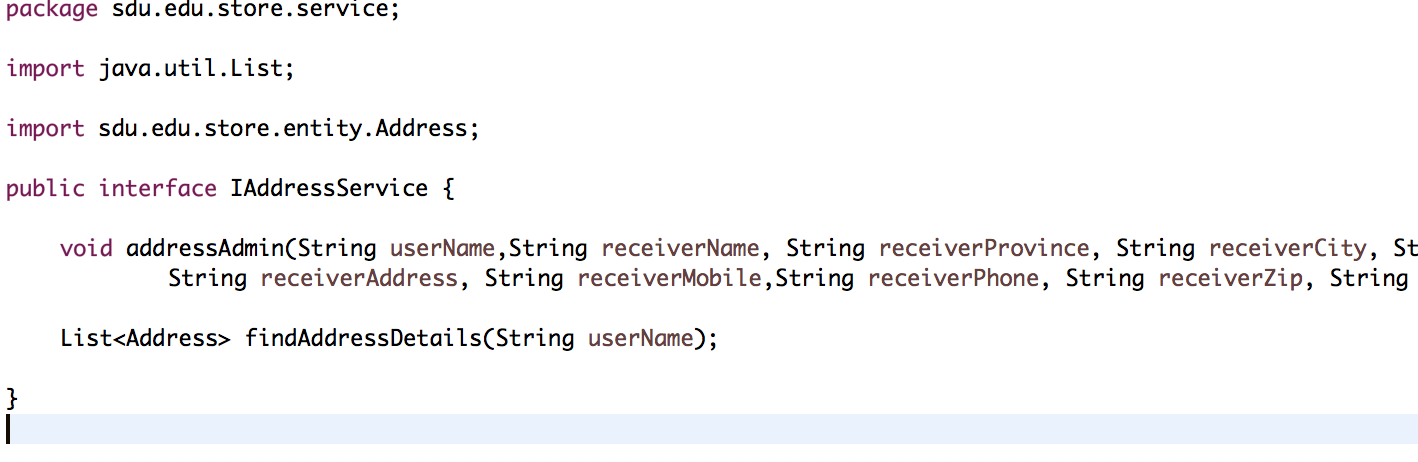
**return** addressMapper.getAddressDetails(userName);

}

## 4.接口

1. 业务接口

List<Address> findAddressDetails(String userName);



(2)Mapper接口

AddressMapper.java

List<Address> getAddressDetails(String userName);



## 5.Mapper文件

<select id=*"getAddressDetails"* resultType=*"sdu.edu.store.entity.Address"*>

select

id,

uid,

recv\_name,

recv\_province,

recv\_city,

recv\_area,

recv\_district,

recv\_addr,

recv\_phone,

recv\_tel,

recv\_zip,

recv\_tag,

created\_user,

created\_time,

modified\_user,

modified\_time

from

t\_address

where

created\_user=#{userName}

</select>

## 6.数据库表

user表

