**校园失物招领平台**

# 1. 项目简介

本系统为学生更好地找回遗失物品提供了一个信息交换的共享平台，用户可以在平台上查看其它用户捡到的失物，并且根据自己失物的信息来在失物库中进行查找。此外，本应用还提供注册和登录账号的功能。

具体功能：

（1）实现了账号登录、注册、修改密码、发布失物信息、查找失物信息五个界面。

（2）实现了对五个界面信息的数据库读取显示及增删改查功能。

# 2 后端API设计

（1）用户注册 Post：<https://localhost:44307/api/User/Register>

调用过程：需要向传递一个User对象，注册成功返回User对象。主体代码如下：先判断有无重名用户，其次将密码进行MD5编码之后存入数据库。

//注册

public Res register(t\_user data) {

Res res = new Res();

//新增一行数据

dbcontext.Add(data);

//保存到数据库

int row = dbcontext.SaveChanges();

if (row > 0)

{

res.code = 1;

res.msg = "注册成功";

}

else {

res.code = -1;

res.msg = "注册失败";

}

return res;

}

//注册

public Res register(User data)

{

Res res = new Res();

//判断请求的数据是否合法

if (string.IsNullOrEmpty(data.name) || string.IsNullOrEmpty(data.pwd))

{

res.code = -1;

res.msg = "账号密码不能为空";

}

else

{

//检查账号是否在用户表里面，在的话就是被注册了

t\_user check = userDal.getUser(data.name);

if (check != null)

{

res.code = -1;

res.msg = "账号已经被注册了，请使用其他账号";

}

else

{

//从接口请求过来的数据赋值给数据库的表对象

t\_user user = new t\_user();

user.name = data.name;

user.nikename = data.nikename;

//用md5加密密码

user.pwd = Md5Help.UserMd5(data.pwd);

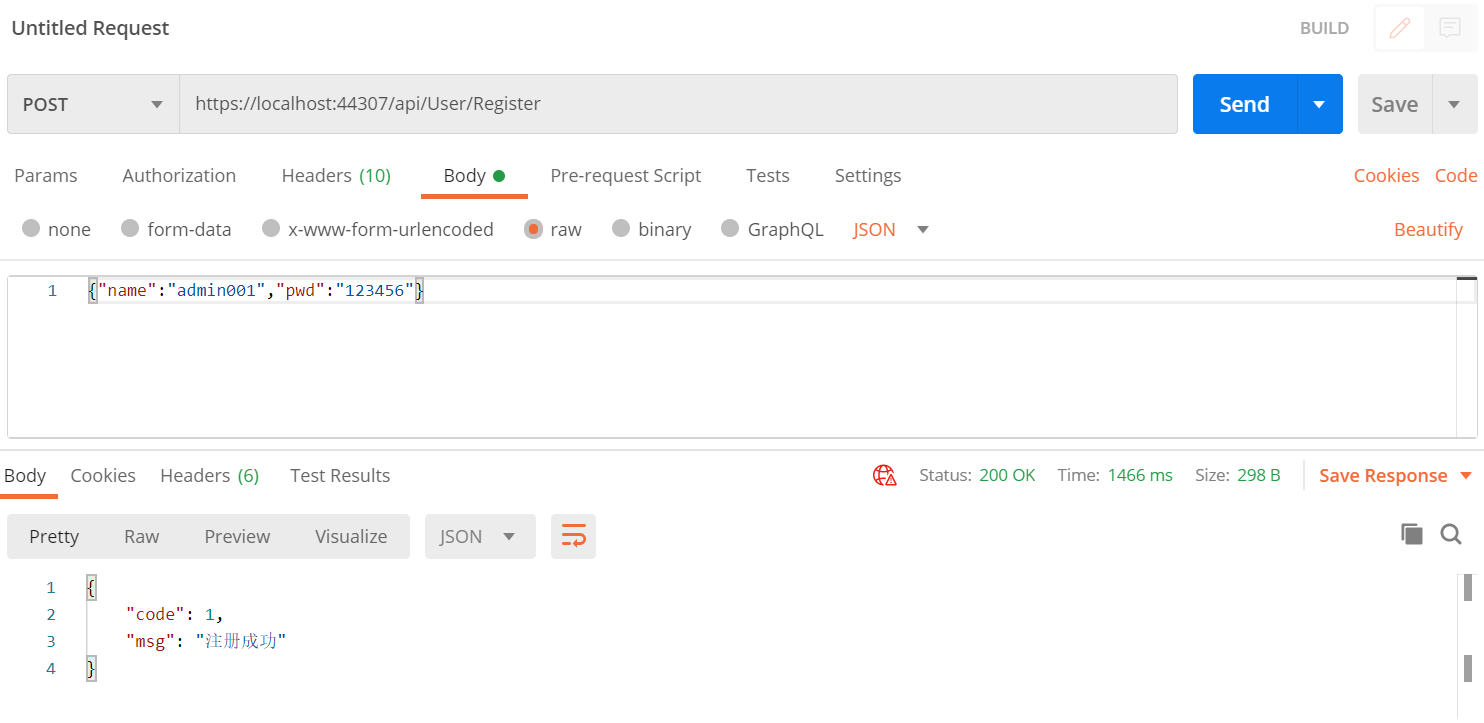
res = userDal.register(user);

}

}

return res;

}



（2）用户登录 post：<https://localhost:44307/api/User/Login>

调用过程：需要向传递一个User对象，先判断此用户存在与否，再判断密码是否匹配，主体过程如下所示。

//登录

public Res login(User data) {

Res res = new Res();

//判断请求的数据是否合法

if (string.IsNullOrEmpty(data.name) || string.IsNullOrEmpty(data.pwd))

{

res.code = -1;

res.msg = "账号密码不能为空";

}

else {

//检查账号是否在用户表里面，在的话就是被注册了

t\_user user = userDal.getUser(data.name);

if (user == null)

{

res.code = -1;

res.msg = "无此账号";

}

else {

//比对密码是否正确

if (user.pwd.Equals(Md5Help.UserMd5(data.pwd)))

{

//输入的旧密码跟数据库密码一致

res.code = 1;

res.msg = "登录成功";

}

else {

res.code = -1;

res.msg = "登录失败，账户密码有误";

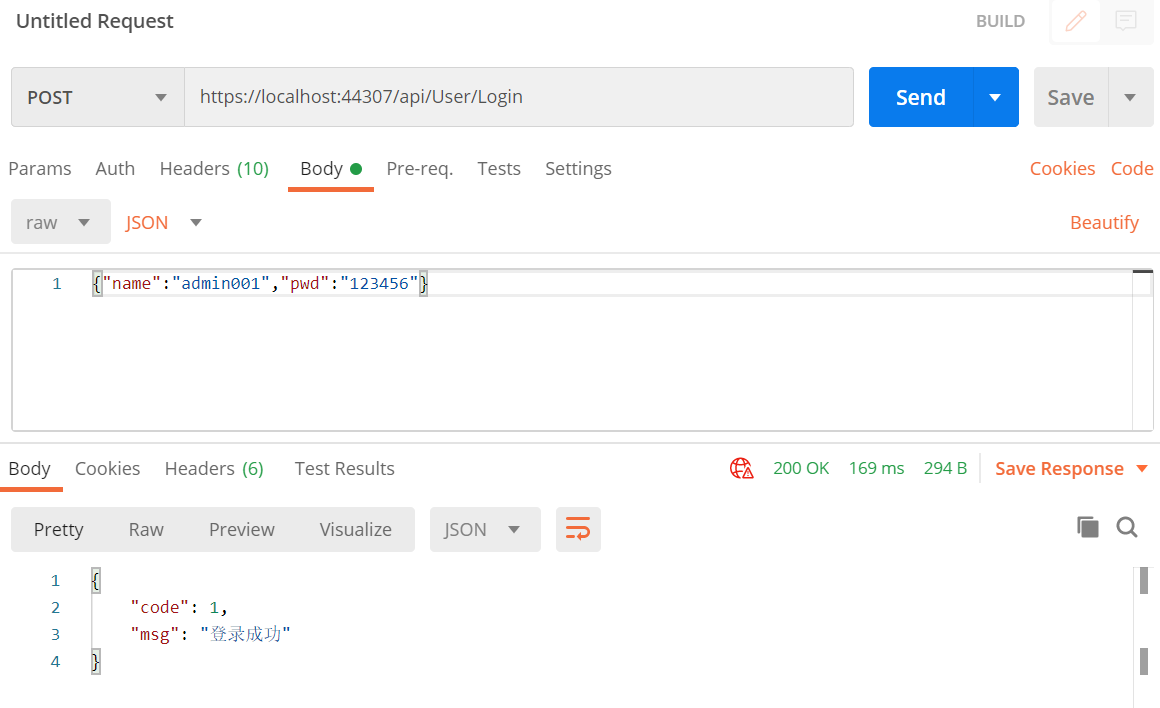
}

}

}

return res;

}



（3）修改密码 Post：<https://localhost:44307/api/User/ModPwd>

调用过程：需要向传递一个UserPwd对象，先判断有无重名用户，其次将旧密码进行MD5编码之后判断该用户的密码是否相符合，相符合则把新密码MD5编码之后存入数据库，否则提示。

//修改用户

public Res modUser(t\_user data) {

Res res = new Res();

//设置修改记录

dbcontext.Update(data);

//保存到数据库

int row = dbcontext.SaveChanges();

if (row > 0)

{

res.code = 1;

res.msg = "修改成功";

}

else

{

res.code = -1;

res.msg = "修改失败";

}

return res;

}

//修改密码

public Res modPwd(UserPwd data)

{

Res res = new Res();

//判断请求的数据是否合法

if (string.IsNullOrEmpty(data.name) || string.IsNullOrEmpty(data.oldpwd) || string.IsNullOrEmpty(data.newpwd))

{

res.code = -1;

res.msg = "账号、旧密码、新密码不能为空";

}

else

{

//检查账号是否在用户表里面，在的话就是被注册了

t\_user user = userDal.getUser(data.name);

if (user == null)

{

res.code = -1;

res.msg = "无此账号";

}

else

{

//比对密码是否正确

if (user.pwd.Equals(Md5Help.UserMd5(data.oldpwd)))

{

//输入的旧密码跟数据库密码一致，修改到数据库

user.pwd = Md5Help.UserMd5(data.newpwd);

res = userDal.modUser(user);

}

else {

//输入的旧密码跟数据库密码不一致，提示

res.code = -1;

res.msg = "输入的旧密码不正确";

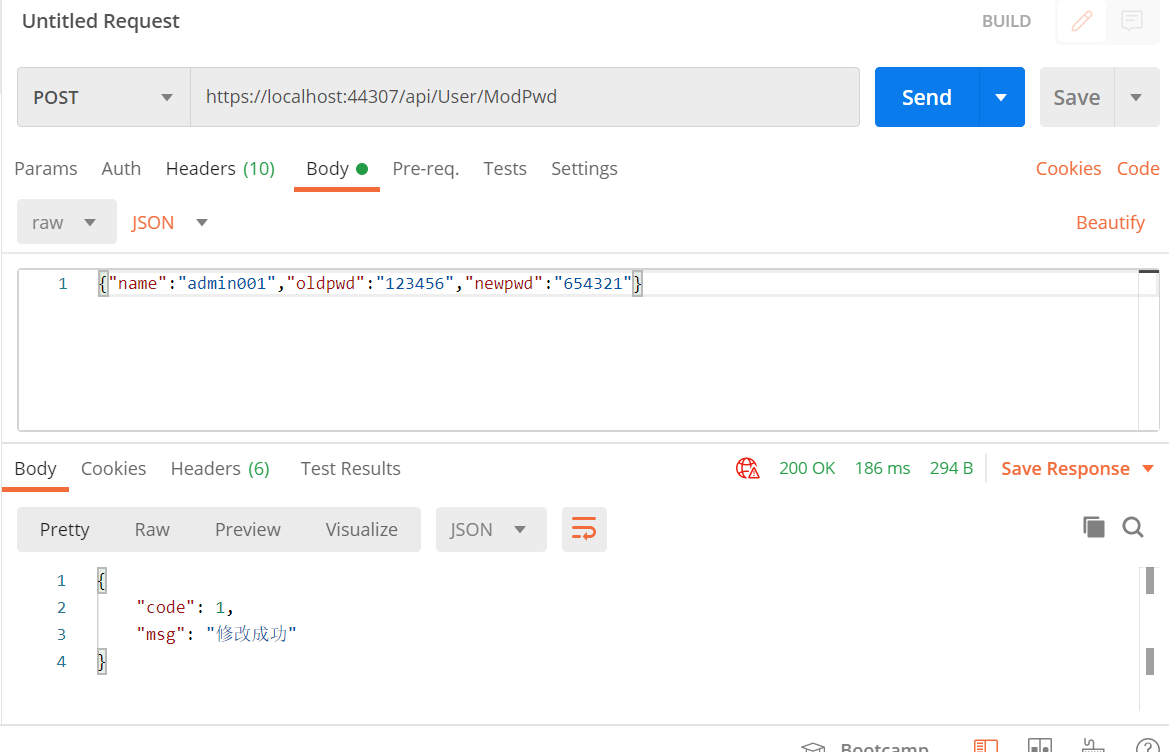
}

}

}

return res;

}



（4）获取招领信息 get：https://localhost:44307/api/Goods/Get

调用过程：可以传递开始日期、结束日期、物品名称、描述，取出符合条件的招领信息返回。

//根据查询条件展示招领信息

public ResData<List<GoodsId>> get(GoodsCon data) {

ResData<List<GoodsId>> res = new ResData<List<GoodsId>>();

List<GoodsId> goods = dbcontext.t\_lostgoods

//查询条件

.Where(t => (t.goods.Contains(data.goods) || data.goods==null) && (t.describe.Contains(data.describe) || data.describe==null) && t.collect\_date>=data.bdate && t.collect\_date<=data.edate)

//查询结果 t\_lostgoods==>GoodsId

.Select(t => new GoodsId(t.id,t.goods,t.describe,t.collect\_date,t.collect\_addr,t.contact\_man,t.contact\_phone,t.contact\_email))

//转换结果

.ToList();

res.code = 1;

res.msg = "查询成功";

res.data = goods;

return res;

}

//根据查询条件展示招领信息

public ResData<List<GoodsId>> get(string goods, string describe, string bdate, string edate)

{

ResData<List<GoodsId>> res = new ResData<List<GoodsId>>();

res.code = 1;

GoodsCon data = new GoodsCon();

//赋值查询条件

if (string.IsNullOrEmpty(goods))

{

data.goods = null;

}

else {

data.goods = goods;

}

if (string.IsNullOrEmpty(describe))

{

data.describe = null;

}

else

{

data.describe = describe;

}

if (string.IsNullOrEmpty(bdate))

{

data.bdate = DateTime.MinValue;

}

else {

try

{

data.bdate = DateTime.Parse(bdate);

}

catch (Exception e) {

res.code = -1;

res.msg = "开始日期不是有效的日期格式";

}

}

if (string.IsNullOrEmpty(edate))

{

data.edate = DateTime.Now;

}

else

{

try

{

data.edate = DateTime.Parse(edate);

}

catch (Exception e)

{

res.code = -1;

res.msg = "结束日期不是有效的日期格式";

}

}

//如果查询条件有误，则返回提示

if (res.code == -1) {

return res;

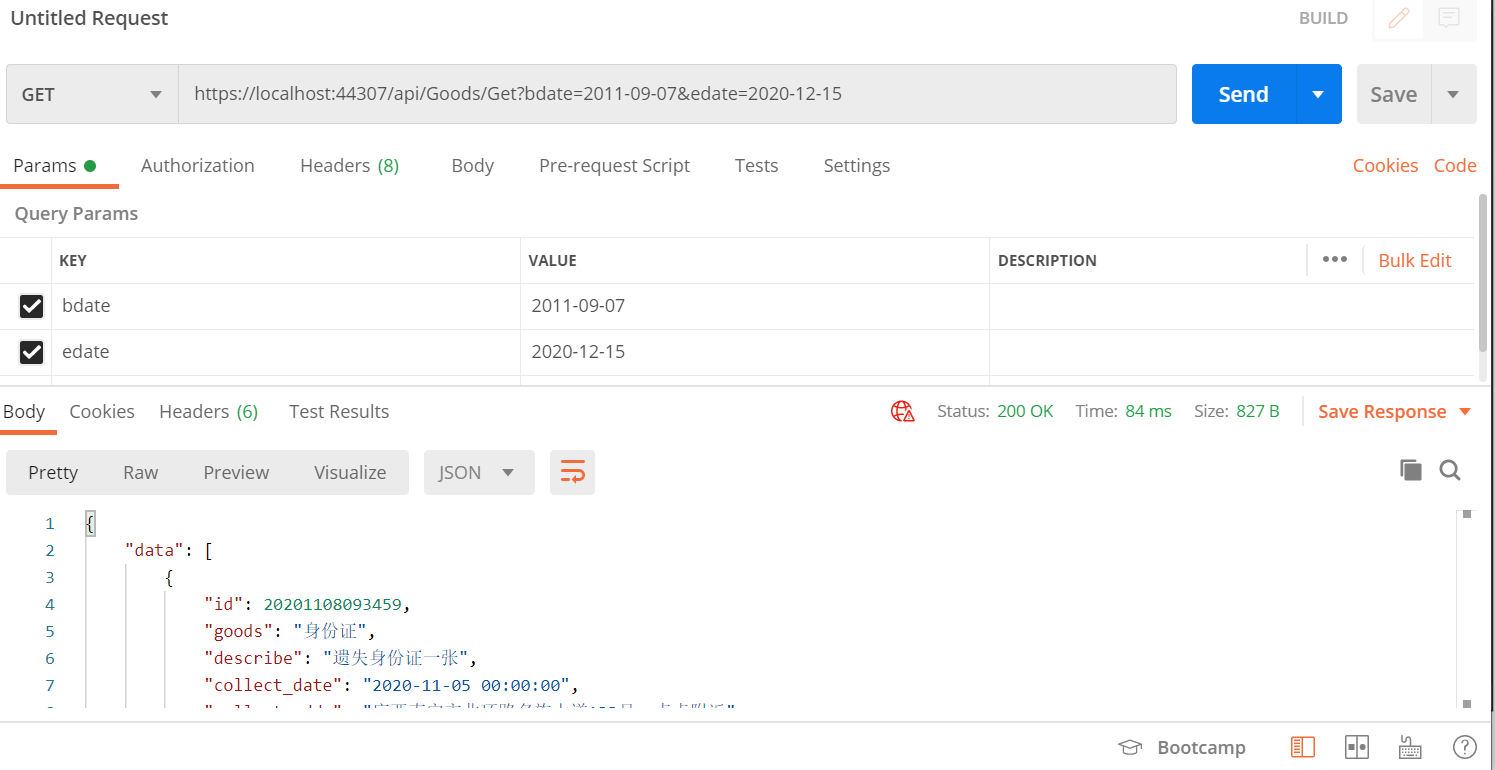
}

//显示查询结果

res = lostGoodsDal.get(data);

return res;

}



（5）发布招领信息 Post：<https://localhost:44307/api/Goods/Release>

调用过程：需要向传递一个Goods对象，先判断判断物品名称、描述联系人等是否完整，如果完整则保存到数据库并返回发布成功的信息ID，否则提示。

//发布招领信息

public ResData<Release> releaseGoods(t\_lostgoods data) {

ResData<Release> res = new ResData<Release>();

try

{

//增加一行记录

dbcontext.Add(data);

//保存到数据库

int row = dbcontext.SaveChanges();

if (row > 0)

{

res.code = 1;

res.msg = "发布成功";

Release release = new Release();

release.id = data.id;

res.data = release;

}

else

{

res.code = -1;

res.msg = "发布失败";

}

}

catch (Exception ex) {

res.code = -1;

res.msg = "发布失败："+ ex.Message;

}

return res;

}

//发布招领信息

public ResData<Release> releaseGoods(Goods data)

{

ResData<Release> res = new ResData<Release>();

//判断请求的数据是否合法

if (string.IsNullOrEmpty(data.goods)

|| string.IsNullOrEmpty(data.describe)

|| string.IsNullOrEmpty(data.contact\_man))

{

res.code = -1;

res.msg = "物品名称、物品描述、联系人不能为空";

}

else if (string.IsNullOrEmpty(data.contact\_phone) && string.IsNullOrEmpty(data.contact\_email))

{

res.code = -1;

res.msg = "必须输入一种联系方式（手机号码/EMAIL）";

}

else

{

//从接口请求过来的数据赋值给数据库的表对象

t\_lostgoods lostgoods = new t\_lostgoods();

lostgoods.id = long.Parse(DateTime.Now.ToString("yyyyMMddhhmmss"));

lostgoods.goods = data.goods;

lostgoods.describe = data.describe;

lostgoods.collect\_date = data.collect\_date;

lostgoods.collect\_addr = data.collect\_addr;

lostgoods.contact\_man = data.contact\_man;

lostgoods.contact\_phone = data.contact\_phone;

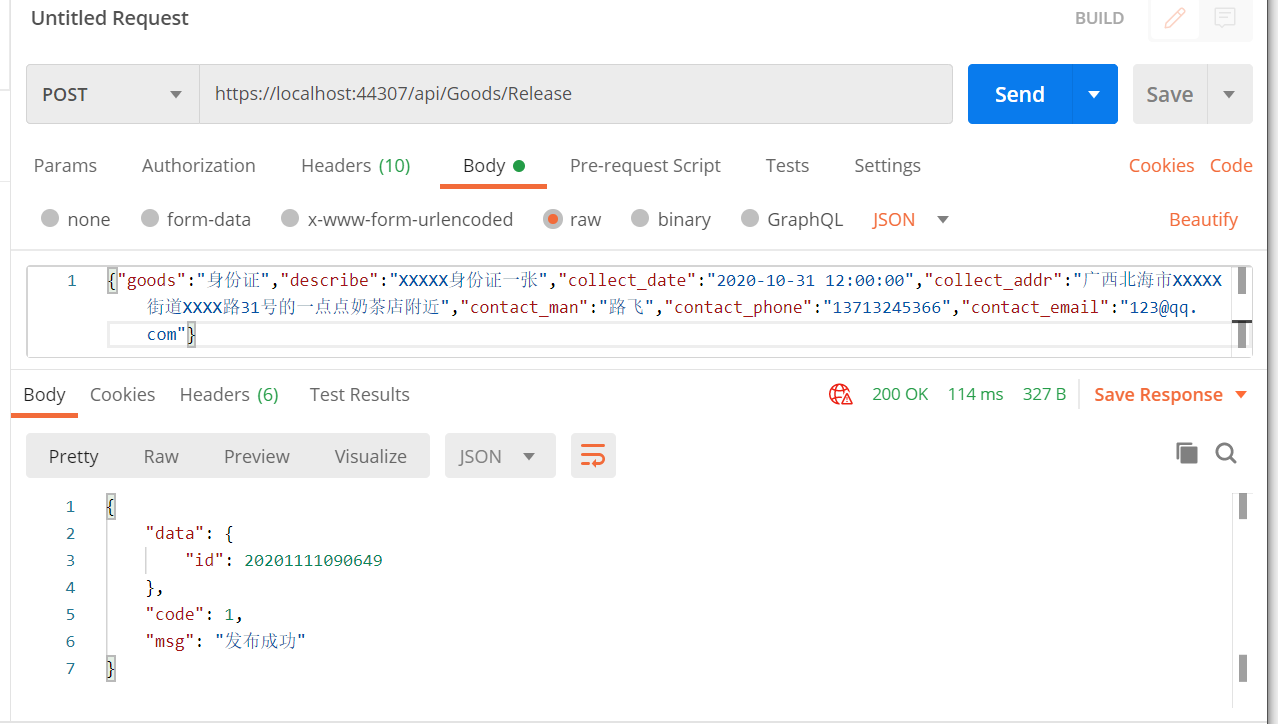
lostgoods.contact\_email = data.contact\_email;

res = lostGoodsDal.releaseGoods(lostgoods);

}

return res;

}

（6）修改招领信息 Post：https://localhost:44307/api/Goods/Mod

调用过程：需要向传递一个GoodsId对象，先判断修改的信息是否完整，不完整则提示，然后判断修改的信息ID是否存在，如果存在则保存相关的信息到数据库，否则提示。

//修改招领信息

public Res modGoods(t\_lostgoods data) {

Res res = new Res();

//设置修改记录

dbcontext.Update(data);

//保存到数据库

int row = dbcontext.SaveChanges();

if (row > 0)

{

res.code = 1;

res.msg = "修改成功";

}

else

{

res.code = -1;

res.msg = "修改失败";

}

return res;

}

//修改招领信息

public Res modGoods(GoodsId data)

{

Res res = new Res();

//判断请求的数据是否合法

if (data.id <= 0)

{

res.code = -1;

res.msg = "ID不能为空";

}

else if (string.IsNullOrEmpty(data.goods)

|| string.IsNullOrEmpty(data.describe)

|| string.IsNullOrEmpty(data.contact\_man))

{

res.code = -1;

res.msg = "物品名称、物品描述、联系人不能为空";

}

else if (string.IsNullOrEmpty(data.contact\_phone) && string.IsNullOrEmpty(data.contact\_email))

{

res.code = -1;

res.msg = "必须输入一种联系方式（手机号码/EMAIL）";

}

else

{

t\_lostgoods lostgoods = lostGoodsDal.getGoods(data.id);

if (lostgoods == null)

{

res.code = -1;

res.msg = "无此招领信息";

}

else

{

//从接口请求过来的数据赋值给数据库的表对象

lostgoods.contact\_man = data.contact\_man;

lostgoods.contact\_email = data.contact\_email;

lostgoods.contact\_phone = data.contact\_phone;

lostgoods.describe = data.describe;

lostgoods.goods = data.goods;

lostgoods.collect\_addr = data.collect\_addr;

lostgoods.collect\_date = data.collect\_date;

//修改到数据

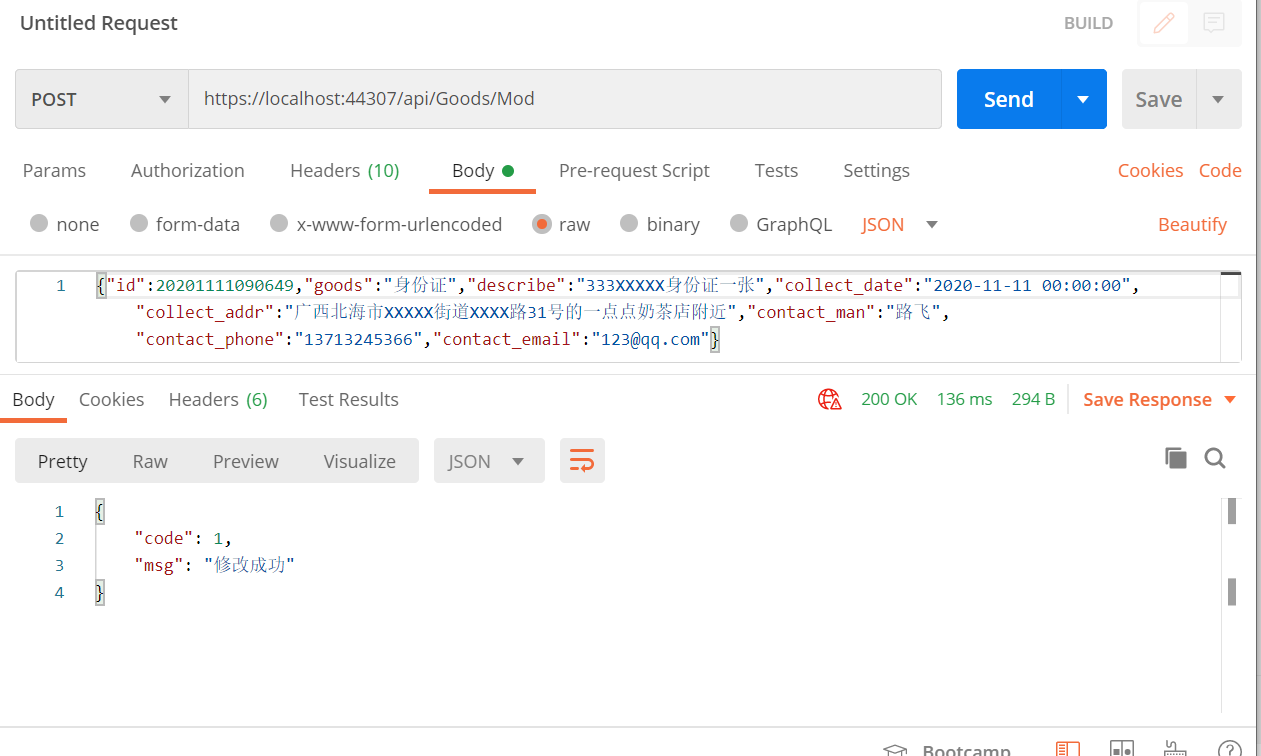
res = lostGoodsDal.modGoods(lostgoods);

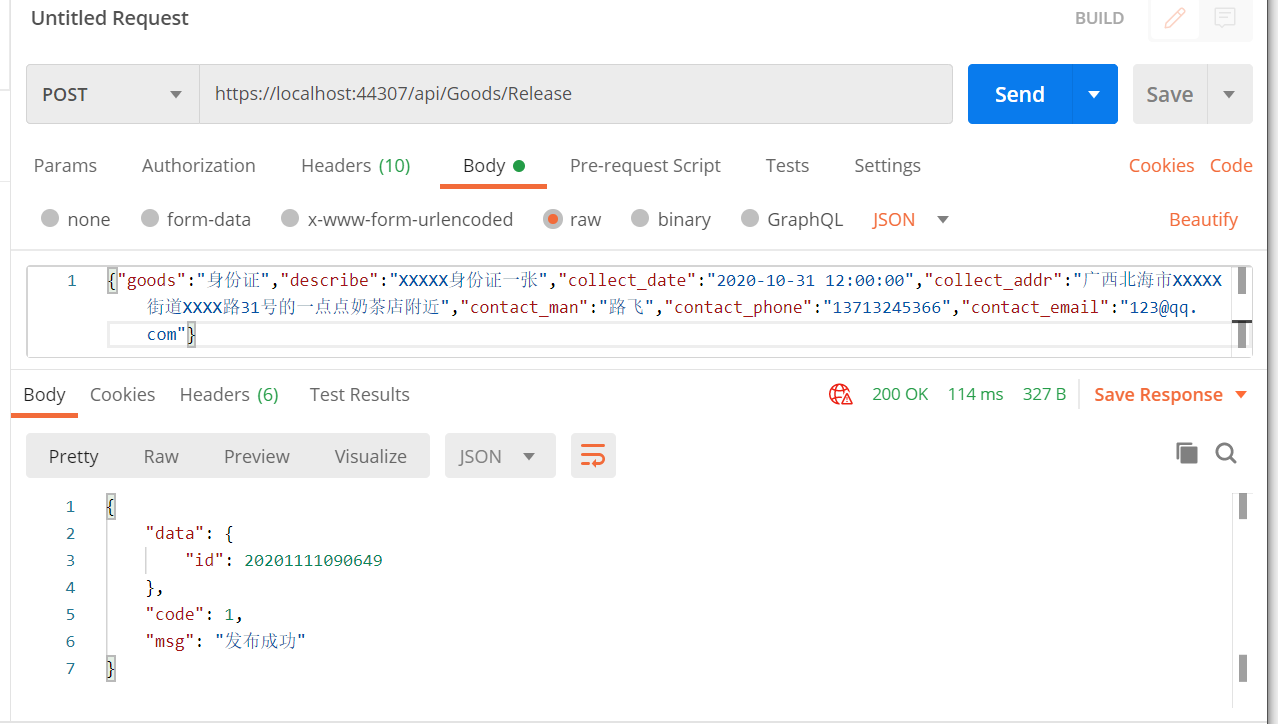
}

}

return res;

}



（7）删除招领信息 Post：https://localhost:44307/api/Goods/Del

调用过程：需要向传递一个Id对象，先判断删除的信息ID是否存在，如果存在则从数据库中删除招领信息，否则提示。

//删除招领信息

public Res delGoods(t\_lostgoods data) {

Res res = new Res();

//设置删除记录

dbcontext.Remove(data);

//保存到数据库

int row = dbcontext.SaveChanges();

if (row > 0)

{

res.code = 1;

res.msg = "删除成功";

}

else

{

res.code = -1;

res.msg = "删除失败";

}

return res;

}

//删除招领信息

public Res delGoods(long data)

{

Res res = new Res();

//判断请求的数据是否合法

if (data <= 0)

{

res.code = -1;

res.msg = "ID不能为空";

}

else

{

t\_lostgoods lostgoods = lostGoodsDal.getGoods(data);

if (lostgoods == null)

{

res.code = -1;

res.msg = "无此招领信息";

}

else

{

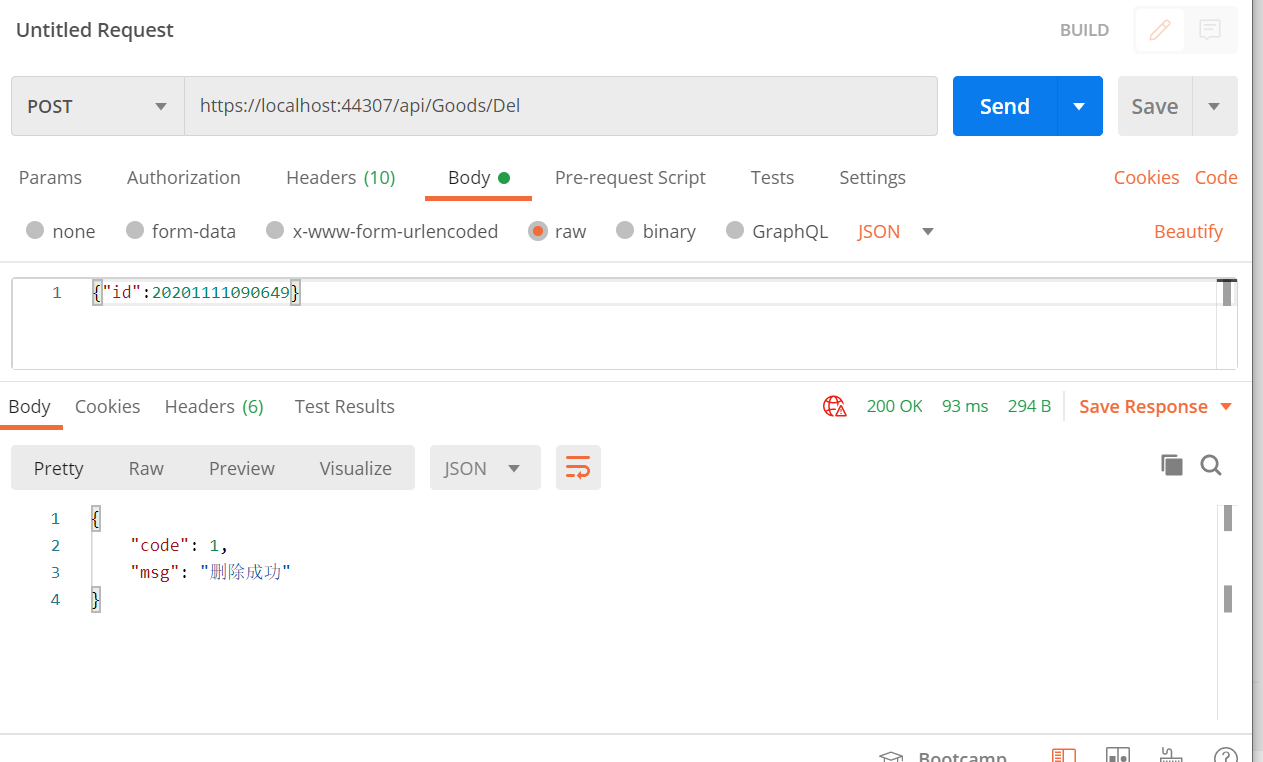
res = lostGoodsDal.delGoods(lostgoods);

}

}

return res;

}

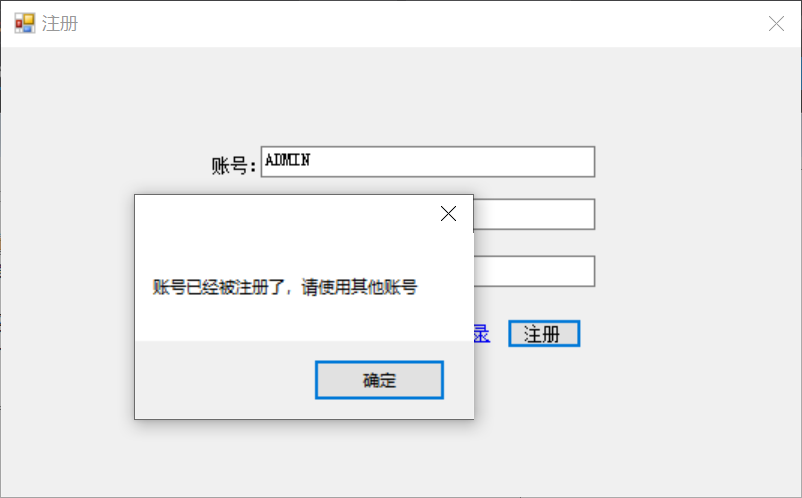


# 3 前端调用

（1）注册

* 请求类型：Http POST
* 方法参数：User
* 返回结果：Res
* URL：<https://localhost:44307/api/User/Register>

调用过程：获取前端UI用户输入的用户名\昵称和密码之后，确认用户名与密码符合规则，之后生成一个User类对象，并将该User类对象转换为Json格式数据通过POST方法发送给后端，取回后端结果后，前端根据结果进行相应展示。注册成功则返回登录界面，注册失败则弹出提示框，效果如图所示：





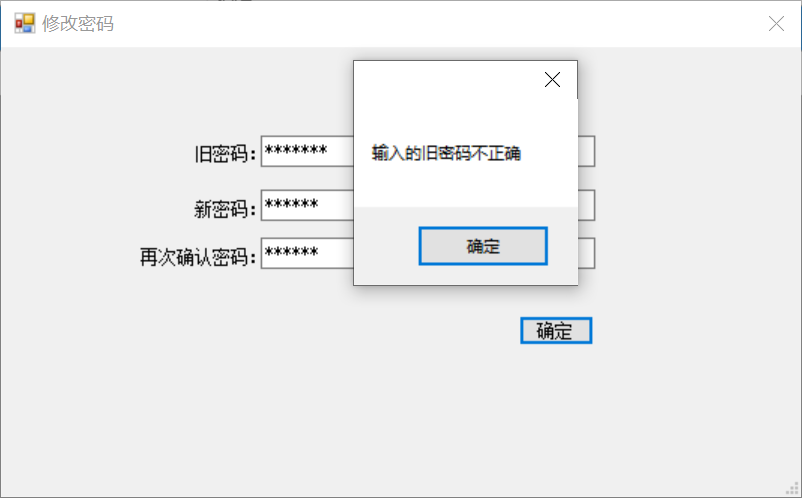
（2）登录

* 请求类型：Http post
* 方法参数：User
* 返回结果：Res
* URL：https://localhost:44307/api/User/Login
* 调用过程：前端界面获取用户输入的用户名以及密码以后，使用这两个数据构造一个单例User对象，然后通过该User类对象调用Login方法，该Login方法会向后端开放的接口发送post方法，参数为User类对象的string类型的name以及pwd属性，之后返回一个JSON的结果，前端UI根据结果进行相应的展示。登录成功则直接进入主界面，登录失败弹出提示框，效果如图所示：



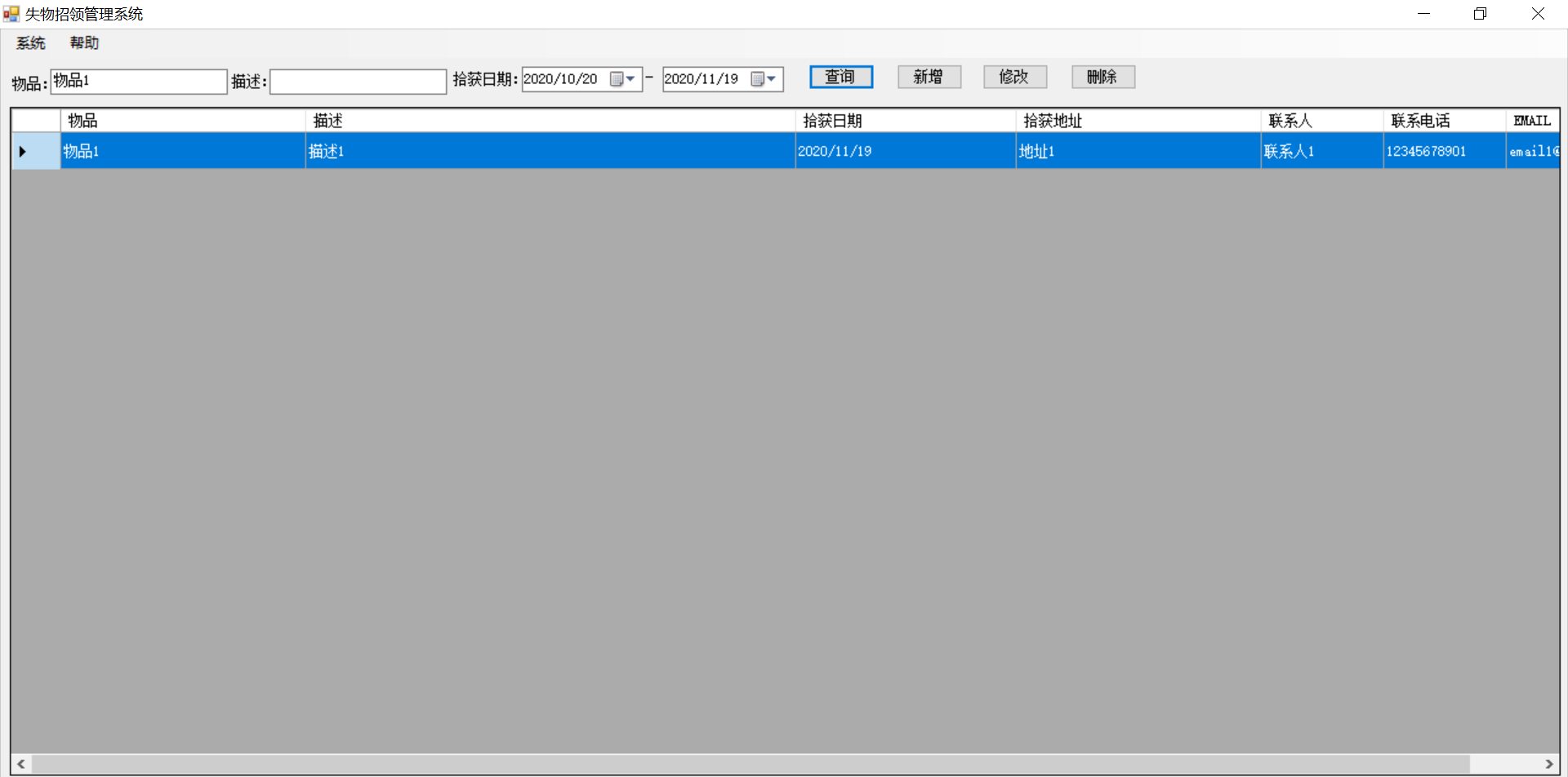
（3）修改密码

* 请求类型：Http post
* 方法参数：UserPwd
* 返回结果：Res
* URL：https://localhost:44307/api/User/ModPwd
* 调用过程：用登录的用户的账号信息、旧密码、新密码一个单例UserPwd 对象，然后通过该UserPwd 类对象调用ModPwd方法，该ModPwd方法会向后端开放的接口发送post方法，之后返回一个JSON的结果，前端UI根据结果进行相应的展示。前端调用效果如图所示：



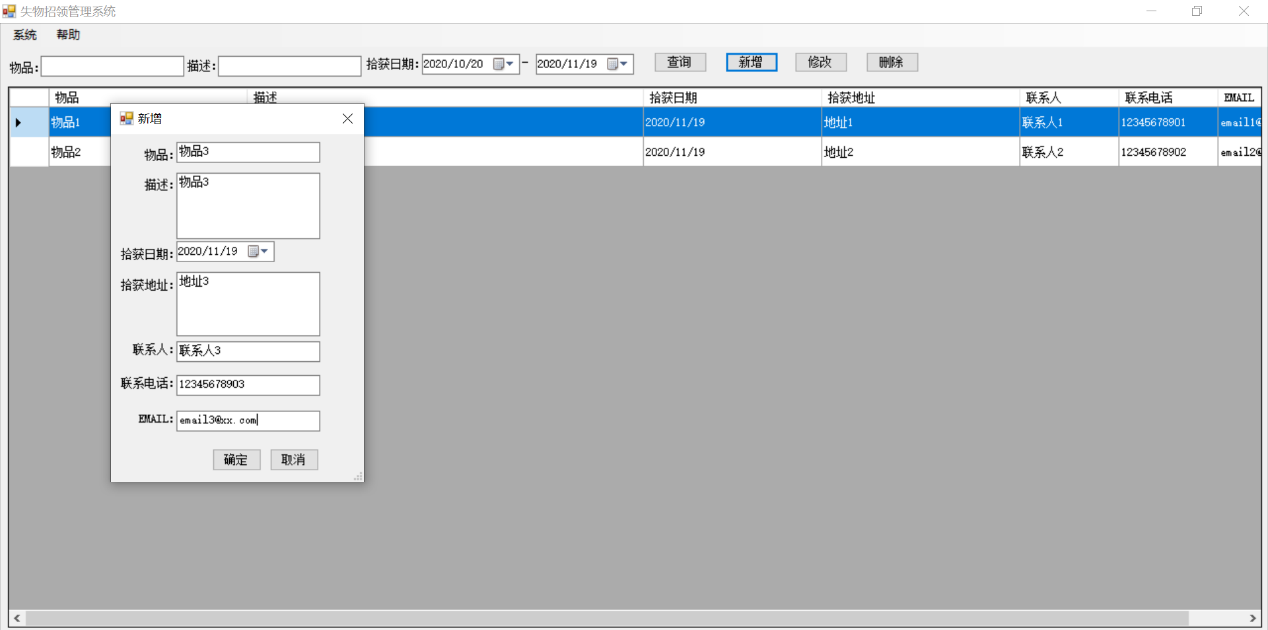
（4）获取招领信息

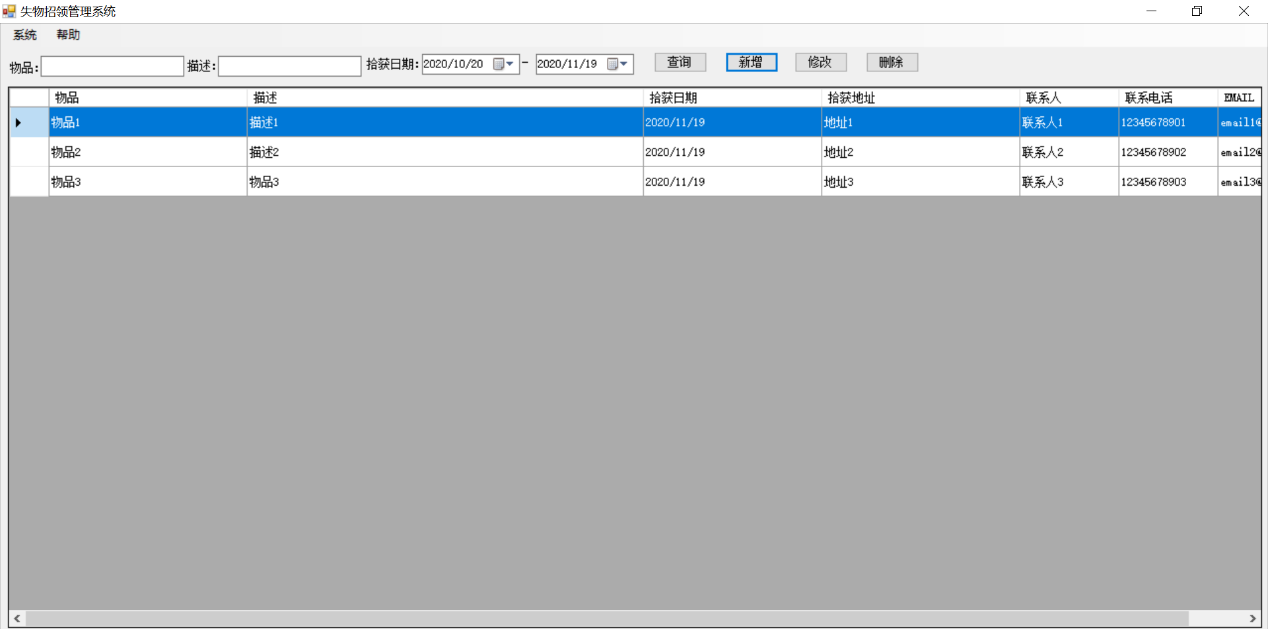
* 请求类型：Http get
* 方法参数：string goods,string describe,string bdate,string edate
* 返回结果：ResData<List<GoodsId>>
* URL：https://localhost:44307/api/Goods/Get
* 调用过程：前端UI根据用户输入的物品名称、物品描述、拾获开始日期和截至日期，会作为URL中的参数，发送get请求给后端，然后根据后端返回的状态码及其list信息进行相应的前端展示。



（5）发布招领信息

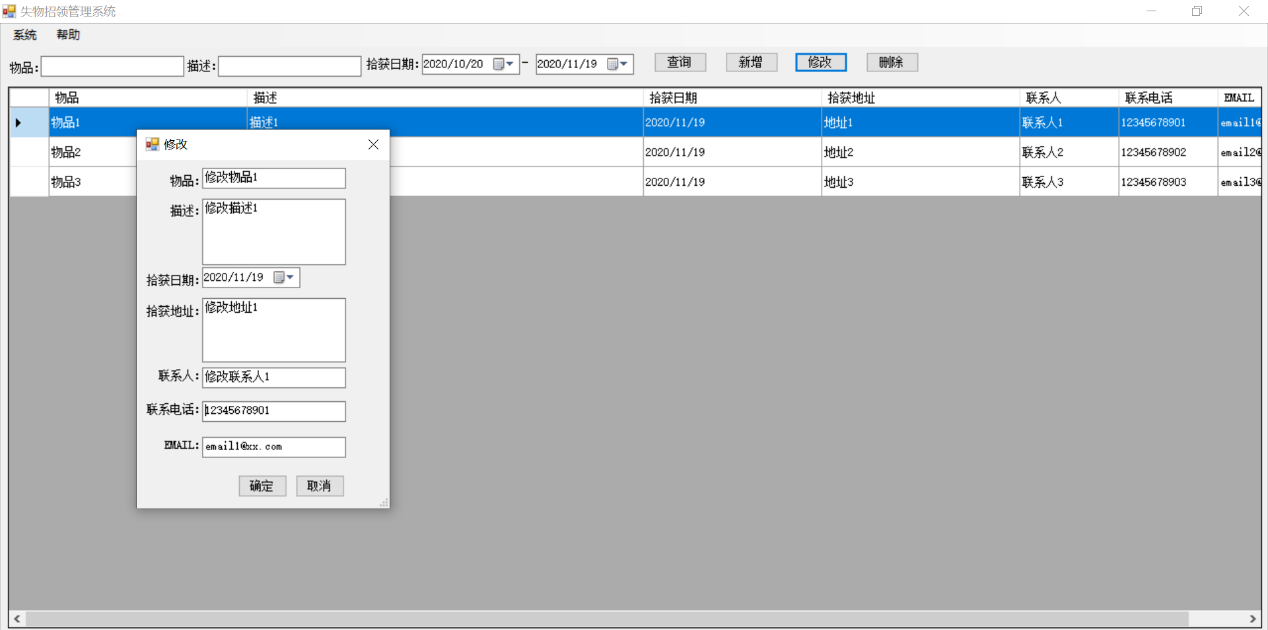
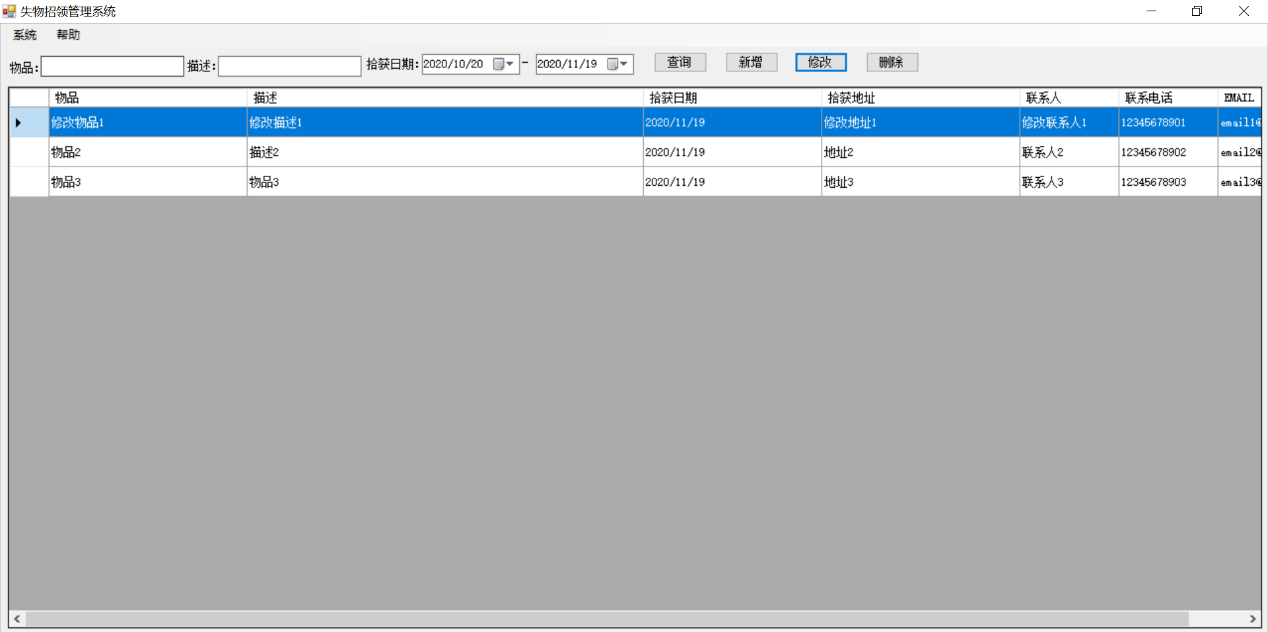
* 请求类型：Http post
* 方法参数：Goods
* 返回结果：ResData<Release>
* URL：https://localhost:44307/api/Goods/Release
* 调用过程：前端UI要求用户输入物品名称、物品描述、联系等信息，之后生成一个Goods类对象，并将该Goods类对象转换为Json格式数据通过POST方法发送给后端，取回后端结果后，前端根据结果进行相应展示。



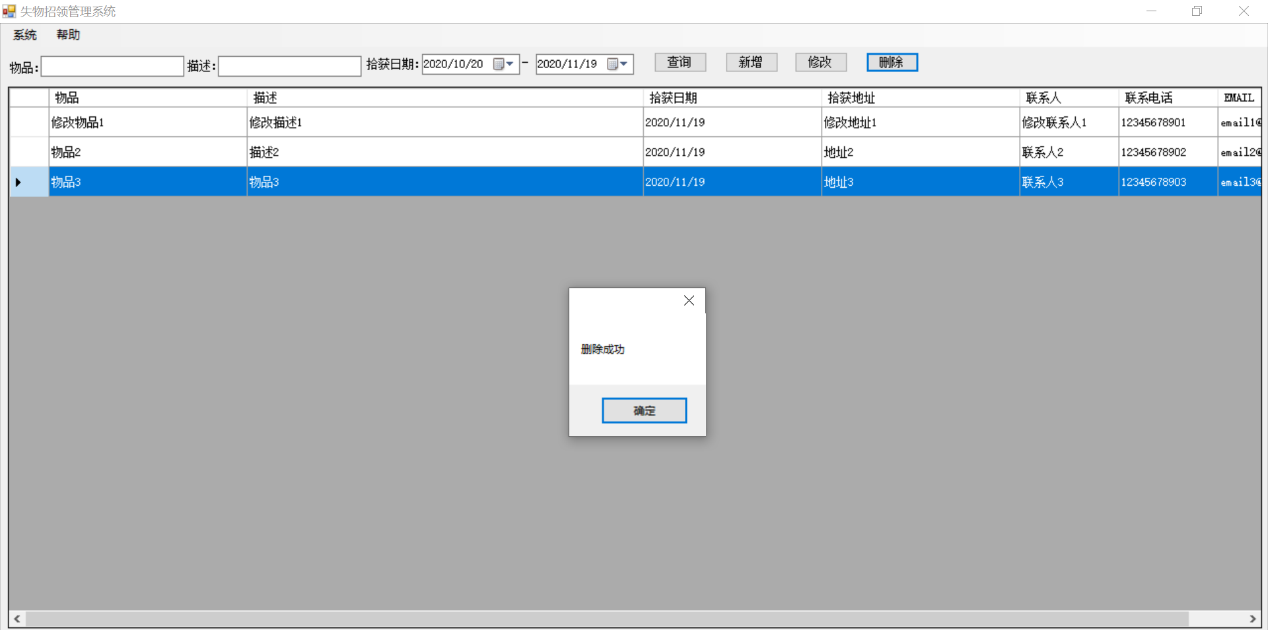


（6）修改招领信息

* 请求类型：Http post
* 方法参数：GoodsId
* 返回结果：Res
* URL：https://localhost:44307/api/Goods/Mod
* 调用过程：前端UI进入某个招领信息的信息维护界面，要求用户输入物品名称、物品描述、联系等信息，之后生成一个GoodsId类对象，并将该GoodsId类对象转换为Json格式数据通过POST方法发送给后端，取回后端结果后，前端根据结果进行相应展示



（7）删除招领信息

* 请求类型：Http post
* 方法参数：Id
* 返回结果：Res
* URL：https://localhost:44307/api/Goods/Del
* 调用过程：用户选中需要修改的招领信息，根据选中的信息ID生成一个GoodsId类对象，并将该GoodsId类对象转换为Json格式数据通过POST方法发送给后端，取回后端结果后进行相应展示。

