



[返回首页](#)

芋道源码 —— 知识星球

我是一段不羁的公告！

记得给芬芳这 3 个项目加油，添加一个 STAR 噢。

<https://github.com/YunaiV/SpringBoot-Labs>

<https://github.com/YunaiV/onemall>

<https://github.com/YunaiV/ruoyi-vue-pro>

2018-03-15

[数据库实体设计](#)

数据库实体设计 —— 交易（2.4）之物流信息（同城配送）

芬芳目前正在做一个开源的电商项目，胖友可以 star 下。

<https://gitee.com/zhi jiantianya/onemall>

1. 概述

本文接 [《数据库实体设计 —— 交易（2.2）之物流信息（快递发货）》](#)，主要分享交易模块的[物流信息](#)的[同城配送](#)的数据库实体设计。

基于如下信息，逆向猜测数据库实体：

[有赞微商城的同城配送](#)

[物流 API](#)

【护脸旁白】

笔者非电商行业出身 && 非有赞工程师，所以有错误或不合理的地方，烦请斧正和探讨。

有赞是个各方面都很 NICE 的公司，[推荐](#)。

2. 背景了解

整体参见 [《同城配送使用教程》](#) 文档。

不同于传统的物流发货，同城配送增加了：

1. 允许选择送达时间
2. 更精细维度的地域范围选择

例如，我们常见的外卖、鲜花、药品等等同城配送。

打脸提示：同城配送涉及到第三方配送，笔者无法方便的整体模拟和测试，如有错误或者不准确的地方，欢迎胖友斧正。

2.1 同城配送功能开关

需要打开开关，用户下单才可以自行选择同城配送。

配置界面如下：



2.2 同城配送配置管理

界面如下：



配送区域设置一共有三种情况：

- 1. 不同区域不同配送费 + 半径原区域
- 2. 不同区域不同配送费 + 自定义区域
- 3. 简易版

我们在下面分享具体数据库实体设计时，在详细分享每种情况。

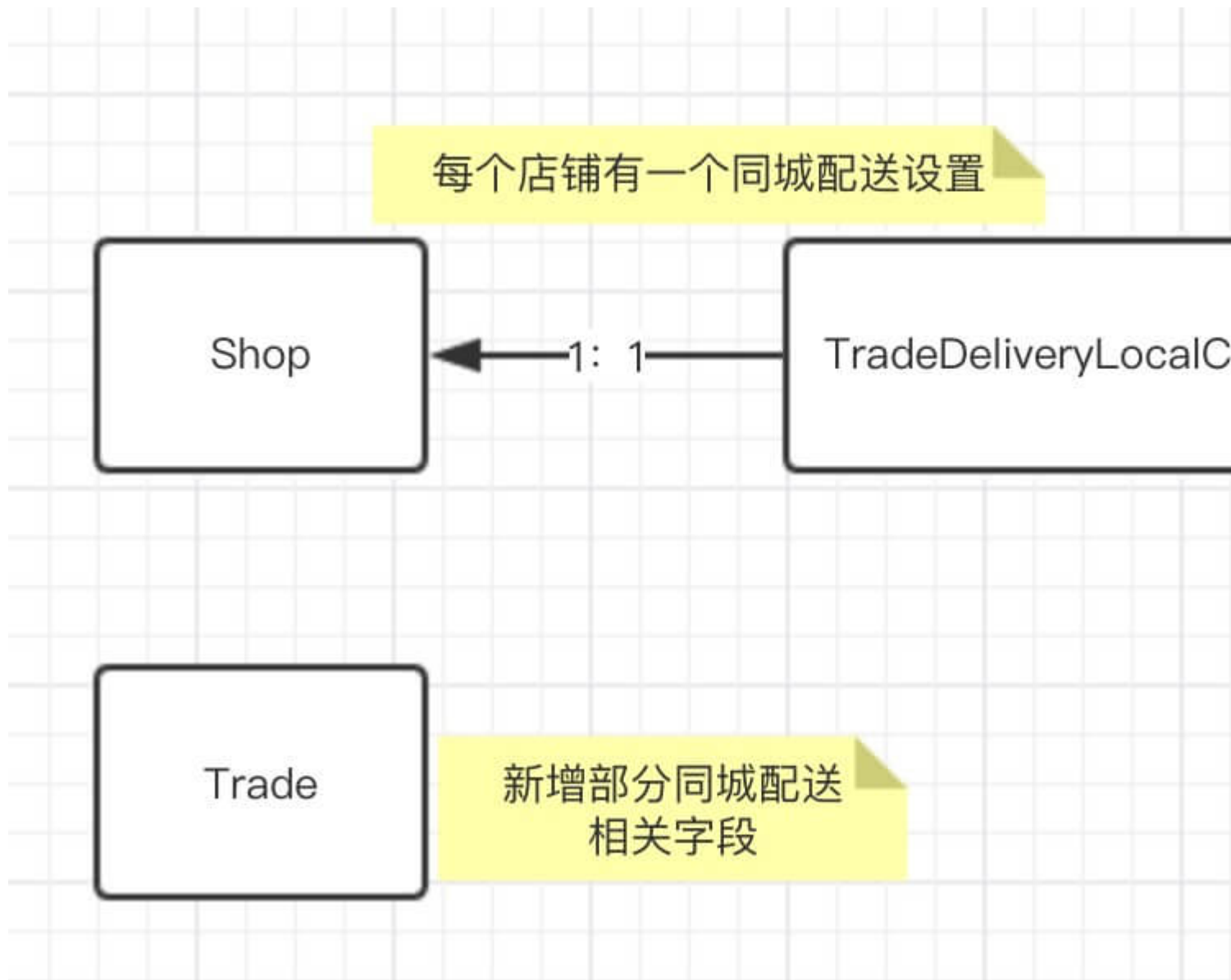
2.3 买家下单选择同城配送

界面如下：



3. 数据库实体

整体实体类关系如下图：



3.1 TradeDeliverySetting

在 [TradeDeliverySetting](#) 中，`isLocal` 标记同城配送功能开关是否开启，`isLocalInTime` 标记同城配送功能的定时达是否开启，在 [《数据库实体设计 —— 交易（2.2）之物流信息（快递发货）》](#) [「3.3 TradeDeliverySetting」](#) 有详细解析。

3.2 TradeDeliveryLocalConfig

[TradeDeliveryLocalConfig](#) ，交易发货同城配送配置。

/**

* 店铺编号 {@link cn.iocoder.doraemon.shopgroup.shop.entity.Shop#id}

```

    */
private Integer id;
/**
 * 创建时间
 */
private Date createTime;
/**
 * 更新时间
 */
private Date updateTime;
/**
 * 配送区域类型
 *
 * 1- 简易版。
 *      没有超区校验，适合配送范围内地图信息不完善的地区，如新开发区、欠发达地区等。
 * 2- 不同区域不同配送费。
 */
private Integer editionType;
/**
 * 配送区域设置数组
 *
 * 使用 JSON 将 {@link AreaModel} 数组 格式化成字符串
 */
private String areaModels;
/**
 * 是否开启定时达
 */
private Boolean isInTime;
/**
 * 配送时间集合
 *
 * 使用 JSON 将 {@link DateRange} 数组 格式化成字符串
 */
private String timeBuckets;
/**
 * 时段细分类型
 *
 * 1-天
 * 2-上午下午晚上（12:00和18:00为分界点）
 * 3-小时
 * 4-半小时
 */
private Integer timeSpanType;
/**
 * 最小可预约时间提前时间，单位：{@link #aheadMinType}
 */
private Integer aheadMin;
/**
 * 最小可预约时间提前时间单位
 *
 * 0-无需提前
 * 1-天（为自然天，如：提前1天，则不管是凌晨1点还是晚上23点，都只能下明天以后的订单）
 * 2-小时
 * 3-分钟
 */
private Integer aheadMinType;
/**
 * 最大可预约时间提前时间，单位：天。
 *
 * 其中，0 为仅限冬天。

```

```

*/
private Integer aheadMax;

```

id ， 店铺编号，指向关联的店铺。Shop : TradeDeliveryLocalConfig = 1: 1 。
editionType ， 配送区域类型。目前有简易版和不同区域不同配送费两种类型。而不同区域不同配送费也包含两种情况，在 [AreaModel.type](#) 属性枚举体现。

===== 配送区域设置 =====

areaModels ， 配送区域设置数组，使用 JSON 将 [AreaModel](#) 数组格式化成字符串进行存储。

AreaModel ， 代码如下：

```

/**
 * 类型
 *
 * 1-半径园 {@link CircleAreaModel}
 * 2-自定义形状 {@link PointModel}
 * 3-简易版
 */
private Integer type;
/**
 * 起步价。单位：分。
 *
 * 起送价：是优惠券/码和满减优惠抵扣前的商品金额，运费不计入起送价。
 */
private Integer startFee;
/**
 * 配送价。单位：分。
 */
private Integer fixedFee;

// ===== 类型【1】BEGIN =====
/**
 * 半径区域，以某个坐标点为中心。
 */
private CircleAreaModel circleAreaModel;
// ===== 类型【1】END =====

// ===== 类型【2】BEGIN =====
/**
 * 构成自定义形状的坐标点集合。
 *
 * 实际该字段包在 polygon_area_model 里，此处笔者为了减少嵌套层级。胖友实际在用，建议还是套在 polygon_area_model 里。
 */
private List<PointModel> pointModels;
// ===== 类型【1】END =====

// ===== 类型【3】BEGIN =====
/**
 * 配送范围介绍
 */
private String desc;
/**
 * 配送范围图片
 */
private String attachPic;
// ===== 类型【3】END =====

```


- 。第一种情况，[CircleAreaModel](#)，不同区域不同配送费 + 半径原区域。



- circleAreaModel ，半径区域，以某个坐标点为中心。
- 第二种情况，[PointModel](#) ，不同区域不同配送费 + 自定义区域。

配送区域设置：



不同区域不同配送费



简易版 ?

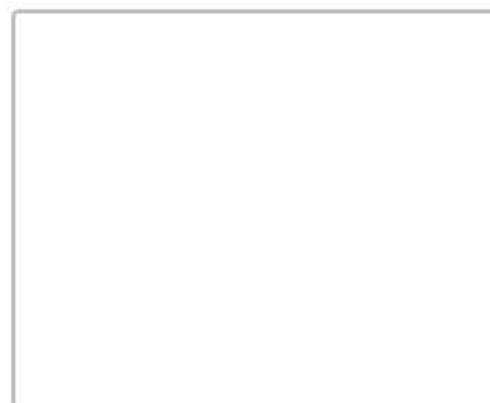


- pointModels ，构成自定义形状的坐标点集合。图中的每个白色空心小点，对应一个 PointModel 对象。通过 PointModel 对象数组，构成自定义形状。
- 第三种情况，简易版。

第三 简易版：没有超区校验，适合配送范围内地图信息
起送价：是优惠券/码和满减优惠抵扣前的商品金额，

配送区域设置： ☐ 不同区域不同配送费 ☒ 简易版 ?

* 配送范围介绍：



配送区域图文信息将会

* 配送范围图片：



建议尺寸：640 x 640 个

* 起送金额：

起送金额为商品促销后

* 配送费：

- 。注意图中弹出的简易版的文字说明。

===== 配送时间设置 =====

isInTime ，是否开启定时达。若关闭该功能，后台无需配置相关时间。另外，买家下单时不能选择配送的时间。

定时达： ☐ 开启定时达功能

开启后，买家下单选择同城送时，需要选择送达时间

此处的配送时间的表单全部

timeBuckets ，配送时间集合，使用 JSON 将 [DateRange](#) 数组格式化字符串进行存储。
timeSpanType ，时段细分类型，分成四种类型。
aheadMin / aheadMinType / aheadMax ，可选择的配送时间范围。

3.3 Trade

我们需要在 [Trade](#) 上，新增同城配送相关的字段。

```
/**
 * 收货地址的纬度
 */
private Double receiverLat;
```

```

/**
 * 收货地址的经度
 */
private Double receiverLng;
/**
 * 同城送订单送达开始时间
 */
private Date deliveryStartTime;
/**
 * 同城送订单送达结束时间
 */
private Date deliveryEndTime;

```

收货地址经纬度相关。在下单时，根据选择的收货地址生成经纬度。通过该经纬度，判断是否满足配送范围。

- receiverLat ： 纬度。
- receiverLng ： 经度。

同城订单配送时间相关。

- deliveryStartTime ， 开始时间。
- deliveryEndTime ， 结束时间。

4. API

基于如下整理 API 类。

[有赞微商城的同城配送](#)

4.1 TradeDeliveryLocalConfigAPI

[TradeDeliveryLocalConfigAPI](#) ， 交易发货同城配送配置 API 。

```

/**
 * 交易发货同城配送配置 API
 */
public interface TradeDeliveryLocalConfigAPI {

    /**
     * 读取商家同城配置的信息
     *
     * https://www.youzanyun.com/apilist/detail/grou
     *
     * @return 交易发货同城配送配置
     */
    YouzanLogisticsLocalGetResult get();

}

```

666. 彩蛋

笔者对同城配送了解的较少，有任何错误或者不正确的地方，超级希望大佬们斧正。

谢谢。

文章目录

1. [1. 概述](#)
2. [2. 背景了解](#)
 1. [2.1. 同城配送功能开关](#)
 2. [2.2. 同城配送配置管理](#)
 3. [2.3. 买家下单选择同城配送](#)
3. [3. 数据库实体](#)
 1. [3.1. TradeDeliverySetting](#)
 2. [3.2. TradeDeliveryLocalConfig](#)
 3. [3.3. Trade](#)
4. [4. API](#)
 1. [4.1. TradeDeliveryLocalConfigAPI](#)
5. [5. 666. 彩蛋](#)

2014 - 2023 芋道源码 |
总访客数 次 && 总访问量 次
[回到首页](#)