
回到首页

芋道源码 —— 知识星球

我是一段不羁的公告！
记得给芳芳这 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
2018-03-05
数据库实体设计

数据库实体设计 —— 交易（2.2）之物流信息（快递发货）

> 芳芳目前正在做一个开源的电商项目，胖友可以 star 下。
> https://gitee.com/zhijiantianya/onemall

1. 概述

本文主要分享交易模块的物流信息的数据库实体设计。

基于如下信息，逆向猜测数据库实体：

> 有赞云提供的物流API
> 有赞微商城的快递发货
>
> 【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

2.1 物流设置

订单支持三种物流方式：

1. 上门自提
2. 同城配送功能
3. 快递发货

通过物流设置，配置三种物流的开关以及全局设置。

本文仅分享快递发货，其他物流方式，在如下文章分享：

1. 《数据库实体设计 —— 交易（2.4）之物流信息（同城配送）》

2. 《数据库实体设计 —— 交易（2.3）之物流信息（上门自提）》

界面如下：

2.2 快递发货运费模板

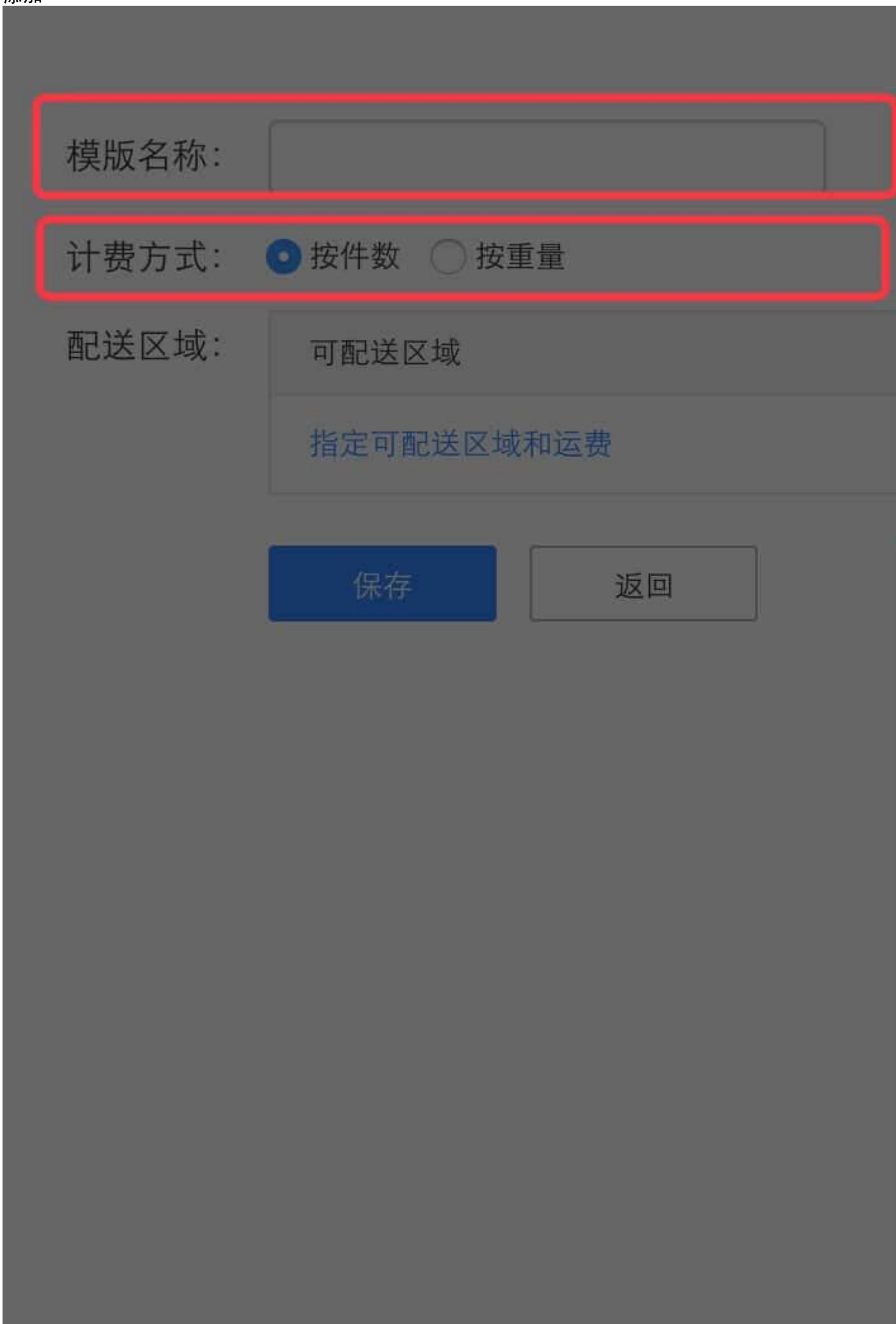
参见文档：

[《如何设置运费模版（按件按重量）？》](#)
[《运费计费规则》](#)

界面如下：

1. 列表

2. 添加



2.3 物流单

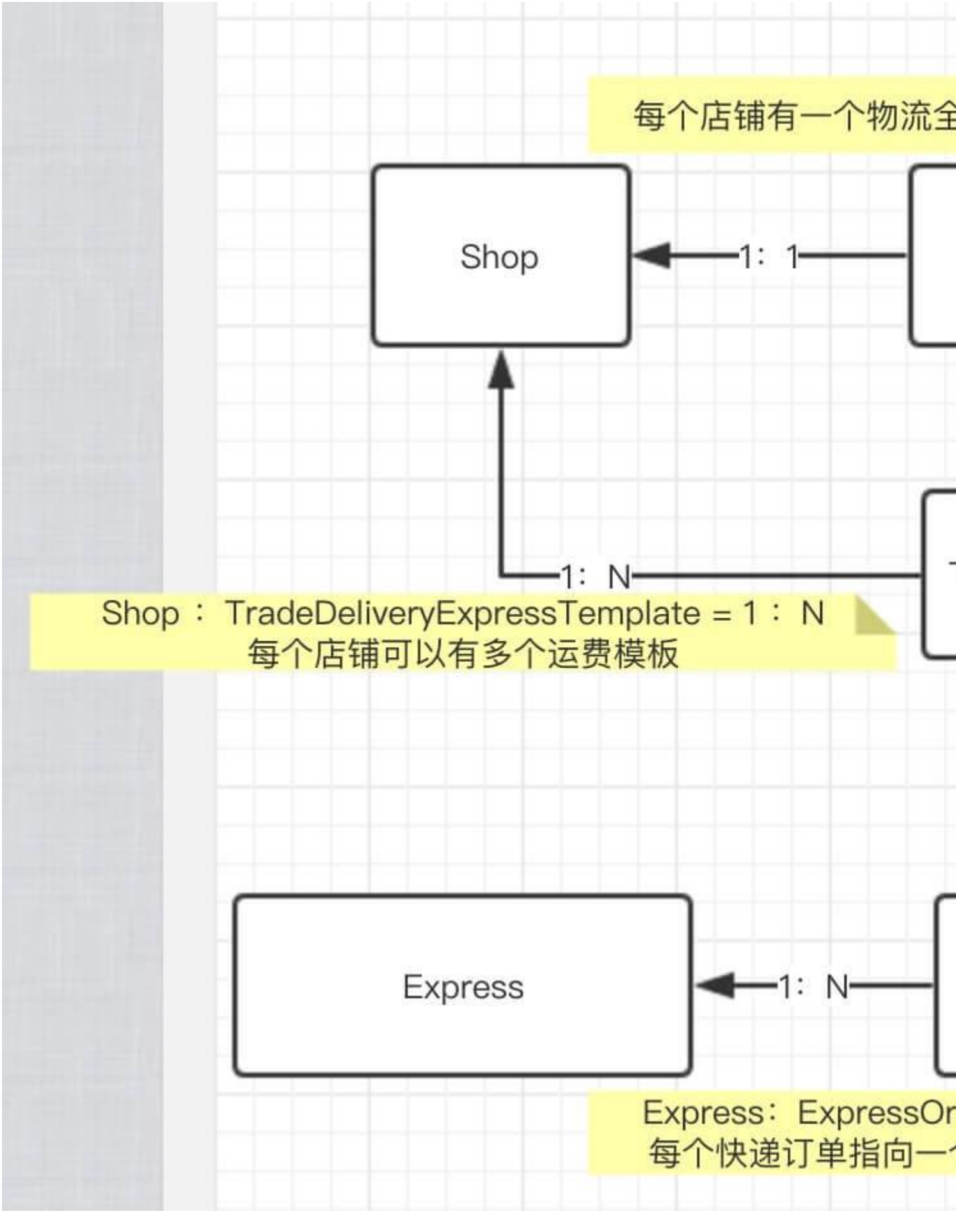
在 《数据库实体设计 —— 交易（2.1）之订单信息》 一文中，我们提到一次交易下，如果购买了多种商品，每种商品会生成一个交易明细。

卖家可以选择多个交易明细进行发货，即一次交易可以有多个物流单。例如，买家购买了 A、B、C 商品，卖家可以按照 [A,B,C] ，也可以 [A,B]+[C]，也可以 [A]+[B]+[C] 以及等等手动发货。

界面如下：

3. 数据库实体

整体实体类关系如下图：



全部实体在 Github 物流实体目录 下可见。

3.1 Express

通用

Express ，快递公司，例如圆通，申通等等。

```
/**
 * 编号
 */
private Integer id;
/**
 * 名字
 */
private String name;
/**
 * 是否展示
 *
 * 1-展示
 * 0-隐藏
 */
private Integer display;
```

数据

胖友可以解析 youzan_express.json ，生成自己的物流公司数据。有如下注意点：

1. 数据来自有赞 API ，感谢有赞。
2. 如果有些物流自己产品暂不支持，可以设置为隐藏。

3.2 ExpressOrder

通用

ExpressOrder ，快递订单。

```
/**
 * 快递单号
 */
private String nu;
/**
 * 快递公司编号 {@link Express#id}
 */
private Integer expressId;
/**
 * 创建时间
 */
private Date createdTime;
/**
 * 快递状态
 *
 * 0：在途，即货物处于运输过程中；
```

```
     * 1：揽件，货物已由快递公司揽收并且产生了第一条跟踪信息；
     * 2：疑难，货物寄送过程出了问题；
     * 3：签收，收件人已签收；
     * 4：退签，即货物由于用户拒签、超区等原因退回，而且发件人已经签收；
     * 5：派件，即快递正在进行同城派件；
     * 6：退回，货物正处于退回发件人的途中；
     */
    private Integer status;
    /**
     * 过程事件数据
     *
     * 使用 JSON 将 {@link Event}数组 格式化成字符串
     */
    private String data;
```

商家发货时，填写好快递单号，会插入记录到该表（当然前提是不存在）。

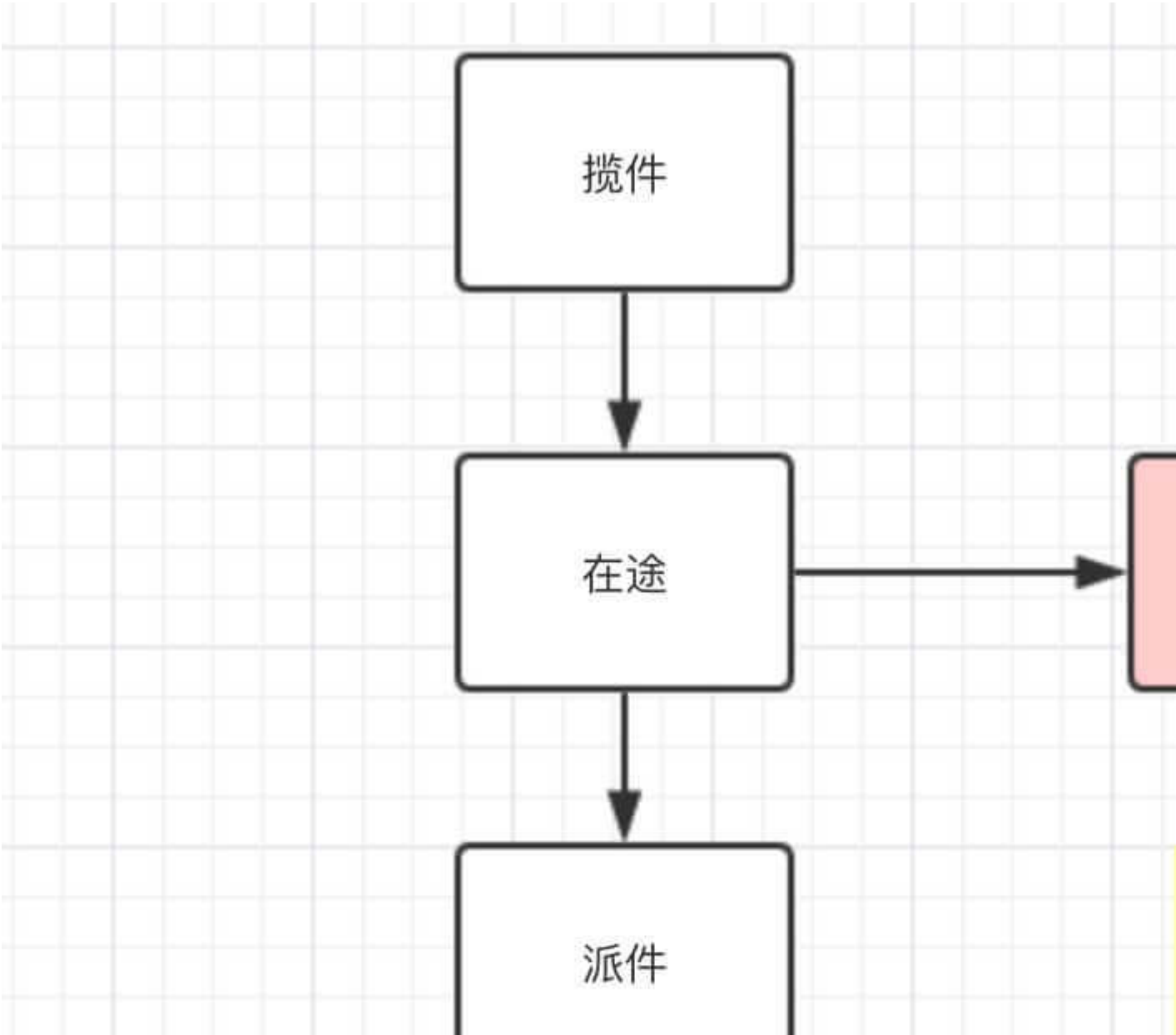
- 「 3.5 TradeDeliveryExpressOrder 」，关联指向 ExpressOrder 。
- 存在单独服务，有定时任务调用不同物流公司的 API 查询快递进度，存储到该表中。猜测有赞使用的是 api.kuaidi100.com 获得物流单号的跟踪信息。

复合主键

- nu ，快递单号。
- expressId ，快递公司编号。

createdTime ，创建时间。注意，这是记录的创建时间，而不是物流订单实际发货的时间。

status ，状态。流转如下图：



`data` ，过程事件数据，使用 JSON 将 Event 数组格式化成字符串。Event 代码如下：

```
public static class Event {

    /**
     * 时间
     */
    private Date time;
    /**
     * 状态
     *
     * 与 {@link ExpressOrder#status} 一致
     */
    private Integer status;
    /**
     * 上下文
     *
     * 例如，到达：上海静安区公司宝山服务部 由 已签收 签收 || 到达：浙江杭州余杭区良渚公司 已揽件
     */
    private String context;

}
```

实际场景下，我们不排除商家乱填或者填错快递单号，系统肯定不能无限轮询 API 查询快递进度。目前笔者的想法是，按照商家填写快递单号，设置一个 `refreshTime` ，超过这个时间 N 天，不再轮询。没太深入思考，胖友有好的思路，欢迎一起讨论起来。

笔者又想了想，ExpressOrder 非业务流程里必须的数据，更多的用处是让用户能方便看到的快递信息。因此，可以考虑在用户进入订单详情页查看快递信息时，调用 api.kuaidi100.com 获得物流单号的跟踪信息。至于是否缓存，笔者觉得不太需要，因为用户不会经常重复看快递信息。当然，胖友的业务场景里需要提供用户物流的进度，例如正在派送中，还是需要存储的。

3.3 TradeDeliverySetting

TradeDeliverySetting ，交易发货设置。

```
/**
 * 店铺编号 {@link cn.iocoder.doraemon.shopgroup.shop.entity.Shop#id}
 */
private Integer id;
/**
 * 更新时间
 */
private Date updateTime;
/**
 * 是否开通快递发货功能
 */
private Boolean isExpress;
/**
 * 计费方式
 *
 * 0-按商品累加运费
 * 1-组合运费（推荐使用）
 */
```

```
private Integer calcType;
/**
 * 是否支持同城
 */
private Boolean isLocal;
/**
 * 是否支持同城定时达
 */
private Boolean isLocalInTime;
/**
 * 是否支持自提
 */
private Boolean isSelf;
```

`id` ，店铺编号。通过该字段，和店铺关联。
`updateTime` ，最后更新时间。
快递发货相关
- `isExpress` ，是否开通快递发货功能。
- `calcType` ，计费方式。目前有两种计费方式，参见 《运费计费规则》 。

同城配送相关
- `isLocal` ，是否支持同城。
- `isLocalInTime` ，是否支持同城定时达。

上门自提相关
- `isSelf` ，是否支持自提。

3.3 TradeDeliveryExpressTemplate

TradeDeliveryExpressTemplate ，交易发货快递运费模板。

```
/**
 * 模板编号
 */
private Integer id;
/**
 * 店铺编号
 */
private Integer shopId;
/**
 * 模板名称
 */
private String name;
/**
 * 状态
 *
 * 1-正常
 * 2-删除
 */
private Integer status;
/**
 * 当前使用次数
 *
 * 当使用次数大于零，不能删除
 */
private Integer useCount;
/**
```

```
     * 付费类型
     *
     * 1-买家付费
     * 2-卖家付费
     */
    private Integer payType;
    /**
     * 是否包邮
     *
     * 0-否
     * 1-是
     * 2-部分
     */
    private Integer freeType;
    /**
     * 复制于哪个模板的编号
     */
    private Integer copyOfTemplateId;
    /**
     * 计算类型
     *
     * 1-按件
     * 2-按重量
     * 3-按体积
     */
    private Integer valuationType;
    /**
     * 运费规则数组
     *
     * 使用 JSON 将 {@link ValueRule}数组 格式化成字符串
     */
    private String valuationRules;
    /**
     * 创建时间
     */
    private Date createTime;
    /**
     * 更新时间
     */
    private Date updateTime;
    /**
     * 删除时间
     */
    private Integer deleteTime;
```

id ， 模板编号。
shopId ， 店铺编号。一个店铺可以有多个模板。
name ， 模板名称。
status ， 状态。删除模板时，标记 status = 2 删除。
useCount ， 当前使用该模板的商品数。当 userCount > 0 时，不允许删除。为什么有该字段？商品模块依赖快递模板，模板删除时，需要保证不存在引用该模板的商品。那么如果没有该字段，删除该模板则需要反向依赖商品模块，计算有多少商品引用了该模板。通过这样的方式，进行解耦。当然，商品引用模板时，需要增长计数；取消引用模板时，需要减小计数。
payType ， 付费类型。目前有赞看到使用卖家付费。
freeType ， 包邮类型。
copyOfTemplateId ， 复制自哪个模板的编号。
valuationType ， 计算运费类型。目前有赞主要使用按件、按重量。

valuationRules ，运费规则数组。使用 JSON 将 ValueRule 数组格式化成字符串进行存储。ValueRule 代码如下：

```
/**
 * 地区编号数组
 */
private List<String> regions;
/**
 * 首件数量，单位根据 {@link #valuationType}
 */
private Integer firstAmount;
/**
 * 首件运费，单位：分
 */
private Integer firstFee;
/**
 * 续件数量，单位根据 {@link #valuationType}
 */
private Integer additionalAmount;
/**
 * 续件运费，单位：分
 */
private Integer additionalFee;
```

- regions ，地区编号数组。存储该字段比较讲究：
  - 选择浙江省下所有的城市，只需要存储浙江省的地区编号。
  - 选择浙江省下部分的城市，存储选择的浙江的城市的地区编号。当然，如果有需要，可以新建一个 ValueRule 对象，配置浙江省剩下部分的城市的价格配置。

这里可能有专业剁手三十年的经验的小伙伴可能就会问：What？包邮怎么处理？

答：包邮在 《满减送》 功能，后续我们分享营销模块时，会专门分享。

3.5 TradeDeliveryExpressOrder

TradeDeliveryExpressOrder ，交易发货快递订单。

```
/**
 * 交易编号 {@link cn.iocoder.doraemon.tradegroup.trade.entity.Trade#id}
 */
private String tid;
/**
 * 交易明细编号 {@link cn.iocoder.doraemon.tradegroup.trade.entity.TradeOrder#id}
 */
private Long oid;
/**
 * 快递单号
 */
private String nu;
/**
 * 快递公司编号
 */
private Integer expressId;
/**
```

```
     * 状态
     *
     * 1-正常
     * 2-删除
     */
    private Integer status;
    /**
     * 发货时间
     */
    private Date createTime;
    /**
     * 删除时间
     */
    private Date deleteTime;
```

指向对应的 TradeOrder

- tid ， 交易编号，即 Trade.tid 。
- oid ， 交易明细编号，即 TradeOrder.id 。

指向对应的 ExpressOrder

- nu ， 快递单号，即 ExpressOrder.nu 。
- expressId ， 快递公司编号，即 Express.id 。

status ， 状态。卖家可能交易明细(交易订单)发货时，填写错了快递单号。此时，可以修改信息。修改时，将原有 TradeDeliveryExpressOrder 记录标记 status = 2 删除，并添加新记录。

4. API

基于如下整理 API 类。

[有赞云提供的物流API]

4.1 ExpressAPI

ExpressAPI ， 快递 API 。

4.2 TradeDeliverySettingAPI

TradeDeliverySettingAPI ，交易发货设置 API 。

```java
/**
 * 交易发货设置 API
 */
public interface TradeDeliverySettingAPI {

    /**
     * 获得交易发货设置
     *
     * https://open.youzan.com/api/oauthentry/youzan
     *
     * @return 交易发货设置
     */
    YouzanLogisticsSettingGetResult get();

    /**
     * 设置开关配置
     *
     * https://www.youzanyun.com/apilist/detail/grou
     *
     * @param shopId 店铺编号
     * @param isExpress 是否支持快递
     * @param calcType 计费类型
     * @param isLocal 是否支持同城
     * @param isSelf 是否支持自提
     * @return 是否成功
     */
    Boolean update(Integer shopId, Boolean isExpress

}
```

4.3 TradeDeliveryExpressTemplateAPI

TradeDeliveryExpressTemplateAPI ，交易发货快递运费模板 API 。

```java
/**
 * 交易发货快递运费模板 API
 */
public interface TradeDeliveryExpressTemplateAPI {

    /**
     * 获取店铺所有物流模板列表
     *
     * https://open.youzan.com/api/oauthentry/youzan
     *
     * @param pageNo 页码
     * @param pageSize 分页值，默认20
     * @return 店铺全部物流模板
     */
    List<YouzanLogisticsTemplateSearchResult.Logisti

    /**
     * 创建物流模板
     *
     * https://www.youzanyun.com/apilist/detail/grou
     *
     * @param name 模板名称
     * @param payType 付费类型
     * @param valuationType 计算类型
     * @param valuationRules 运费规则 json格式，转换成
     */
    void create(String name, Integer payType, Intege

    // update 接口，操作界面有，API 暂未提供。胖友自行 YY

    /**
     * 删除模板
```

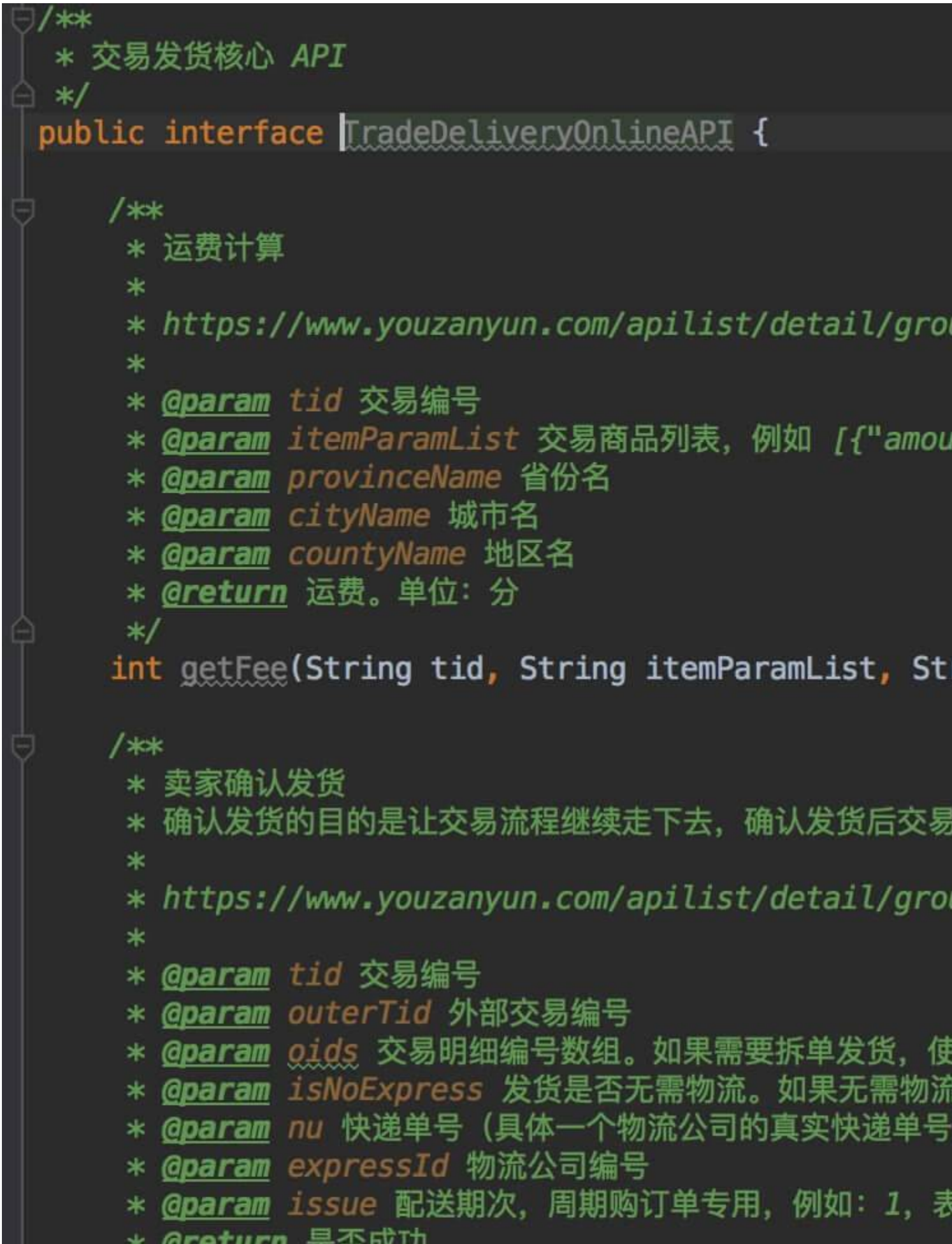
4.4 TradeDeliveryExpressOrderAPI

TradeDeliveryExpressOrderAPI ，交易发货快递订单 API 。

```java
/**
 * 交易发货快递订单 API
 */
public interface TradeDeliveryExpressOrderAPI {

    /**
     * 通过交易号获取所有包裹信息
     *
     * https://www.youzanyun.com/apilist/detail/grou
     *
     * @param tid 交易号
     * @return 物流详情列表
     */
    List<YouzanLogisticsExpressbyordernoSearchResult

    /**
     * 通过交易明细（交易订单）编号获取包裹信息
     *
     * https://www.youzanyun.com/apilist/detail/grou
     *
     * 注意，这个 API 笔者做了调整。
     *       如果传递 tid ，那么和 {@link #search(Strin
     *       并且，返回的只有单条包裹信息，但是一个交易下可
     *       因此，笔者觉得应该是根据交易明细（交易订单）编
     *
     * @param oid 交易明细（交易订单）编号
     * @return 物流详情
     */
    YouzanLogisticsGoodsexpressGetResult get(String

}
```

4.5 TradeDeliveryOnlineAPI

TradeDeliveryOnlineAPI ，交易发货核心 API 。



为什么独立于快递订单的 API？原因如下：

1. 交易有多种物流方式，不仅限于快递。
2. TradeDeliveryExpressOrderAPI 基于 TradeDeliveryExpressOrder 提供API 服务，不提供和 Trade 和 TradeOrder 的关联逻辑。需要关联的逻辑在 TradeDeliveryOnlineAPI 完成。

这块涉及到 API 的拆分，有经验的胖友欢迎一起探讨。

666. 彩蛋

原先把物流想简单的，梳理一次，Get 了蛮多经验的。

下面在了解下上门自提和同城配送功能。思考了下，这两种方式都蛮符合基于医院场景的一些诉求。

文章目录