ETC结算数据监控平台开发文档

（讨论稿）

2020-07

# ETC结算数据监控平台功能需求

详细介绍产品所有的功能需求。

## 省内结算

### 原型图界面





### 需求说明

打开联网结算平台，无需登录即可查看联网结算相关数据图表

加登录页面

#### 基础数据



\*数据更新方式：实时更新与快照

【累计省内交易金额】：2020年起累计产生交易数据中省内交易金额及笔数【实时】

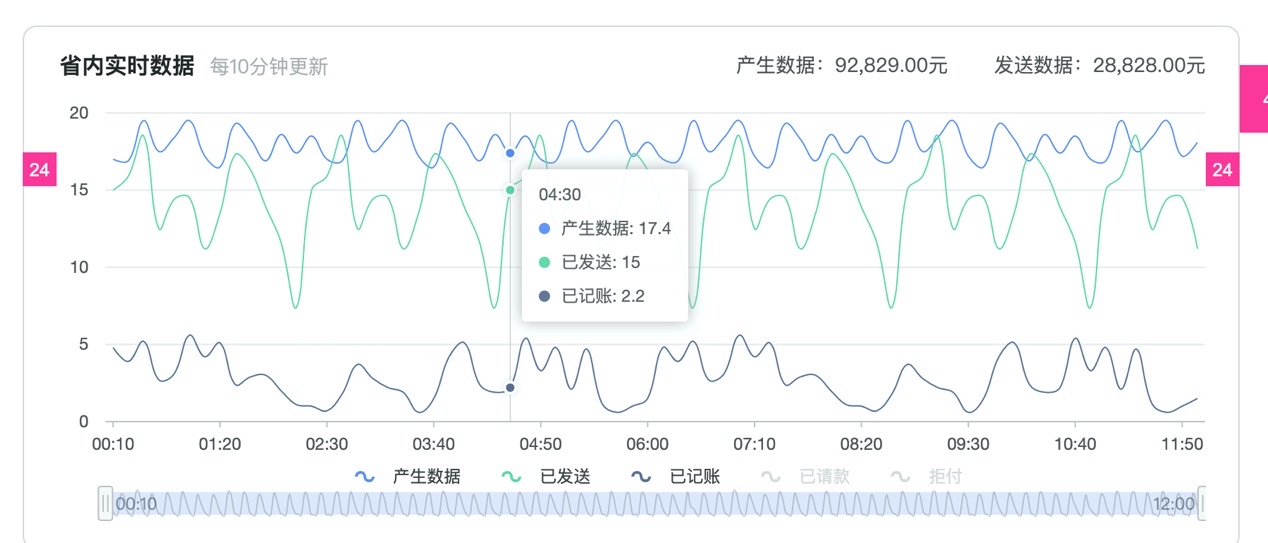
【累计省内发送金额】：2020年起累计发送交易数据中省内交易金额及笔数【实时快照 10分钟】

【累计省内请款金额】：2020年起恒生接口累计确认请款的省内交易金额及笔数【实时快照 10分钟】

【累计（坏账）拒付金额】：2020年起省内交易累计拒收的金额及交易笔数【每天一次】

#### 今日实时数据

\*数据更新方式：每10分钟自动更新



【产生数据】：当日实时产生的省内交易的金额、条数

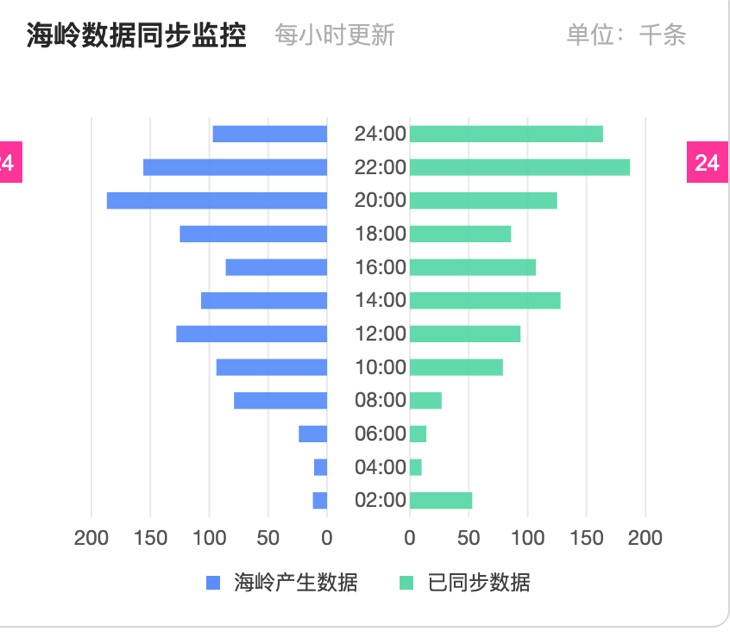
【发送数据】：当日向恒生接口发送交易数据的金额

【记账数据】：当日恒生接口返回的已请款金额

【可视化图表】：每10分钟向数据库中取数据做快照，图表按照快照数据做展示，将产生数据、发送数据、记账数据分别以曲线图展示，支持选择时间分段查看对应时间段的数据

#### 今日消息数量监控

\*数据更新方式：每小时自动更新



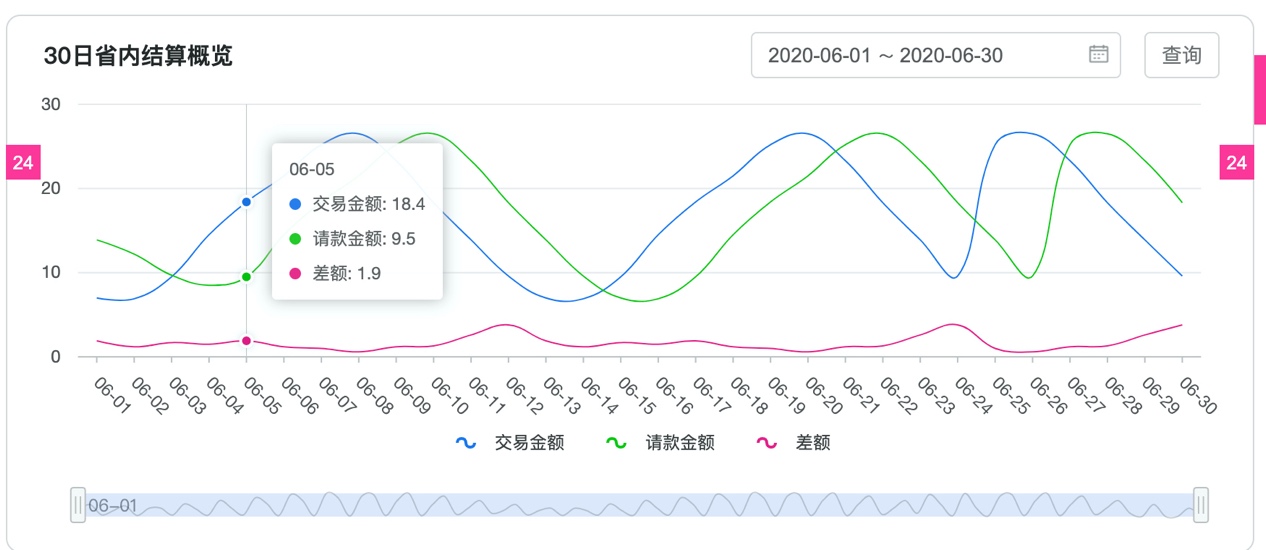
对oracle数量和mysql同步数量按小时做监控：车道出口原始数据、海玲库b\_txf\_chedxfyssj。

同步程序有没有问题

【可视化图表】： 按小时，对oracle数量和mysql同步数据量做快照，并在图标上按小时数据做展示

#### 前30日省内结算数据

\*数据更新方式：每天一次更新统计

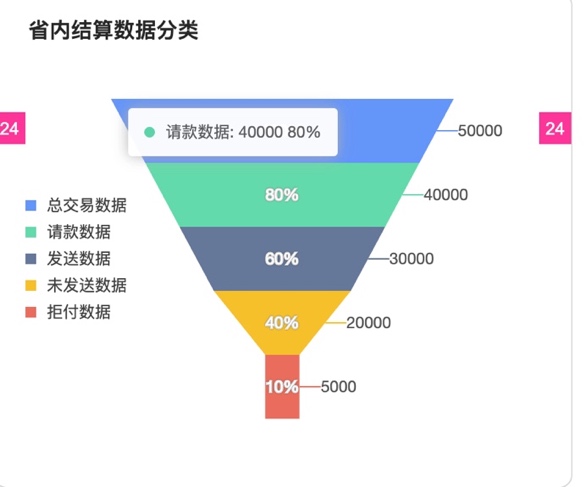


【时间搜索】：默认为截至当前日期前一天的最近30天，支持重新选择时间，范围最大为30天，最少为1天，点击查询后，上方图表根据查询结果更新【所有值】

【可视化图表】：默认展示到前一天的最近30天省内结算数据概览。展示包含当天交易金额、请款金额、差额

#### 省内结算数据状态占比

\*数据更新方式：每次刷新页面读取库中数据占比信息

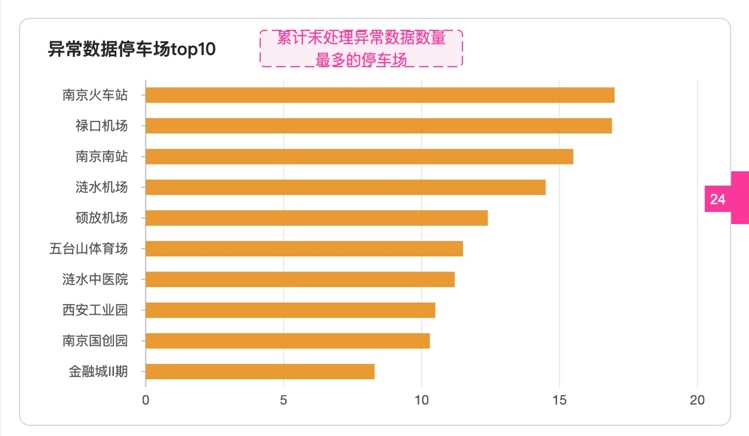


目前数据库中总的省内交易数据条数

根据结算状态按比例展示，包括总交易量，已请款、已记账、拒付（坏账），未发送，已发送等

#### 异常数据停车场top10

\*数据更新方式：每天统计前一天产生异常数据的量

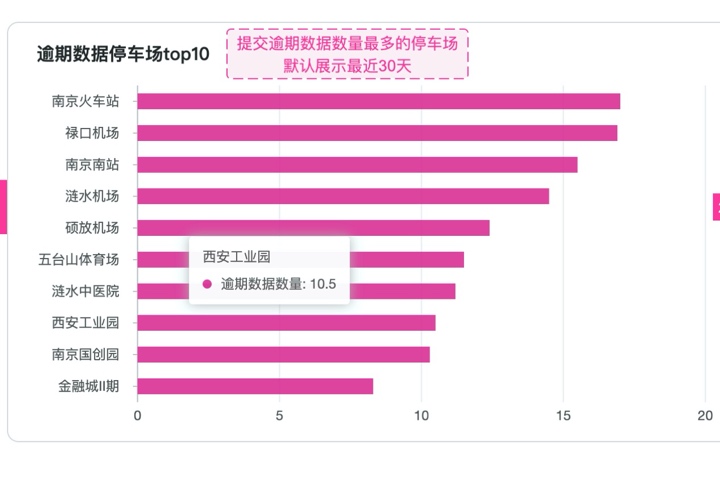


【时间搜索】：默认为截至当前日期前一天的最近30天，支持重新选择时间，范围最大为30天，最少为1天，点击查询后，下方图表根据查询结果更新

【可视化图表】：默认展示到前一天的最近30天产生异常数据数量最多的Top10停车场排行

#### 逾期数据停车场top10【总对总】异常表【逾期状态】

\*数据更新方式：每天统计前一天产生逾期数据的量



【时间搜索】：默认为截至当前日期前一天的最近30天，支持重新选择时间，范围最大为30天，最少为1天，点击查询后，下方图表根据查询结果更新

【可视化图表】：默认展示到前一天的最近30天产生逾期数据数量最多的Top10停车场排行

## 省外结算

### 原型图界面





#### 需求说明

#### 基础数据



\*数据更新方式：每次刷新页面时更新

【累计省外交易金额】：2020年起累计产生省外交易数据交易金额及笔数【实时】

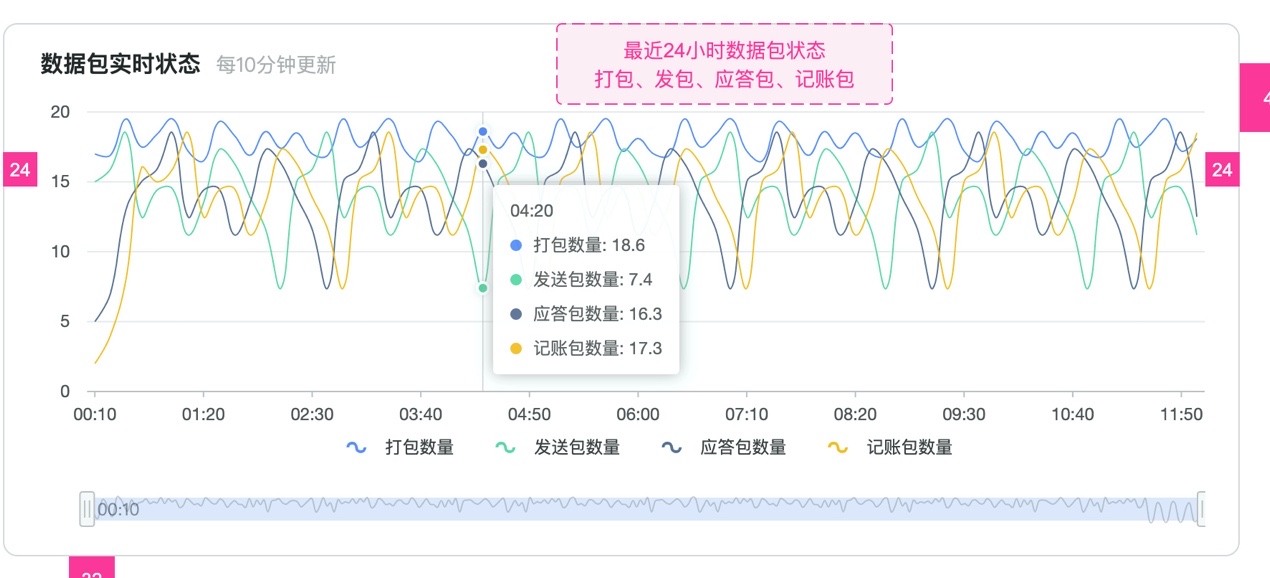
【累计省外清分金额】：2020年起联网中心累计返回清分包中交易金额及笔数【一天一次】

【待处理异常数据】：数据库中标记为未处理的异常数据的金额及笔数。【一天一次】后期增加“数据查询”功能后，点击应跳转至异常数据列表

【待处理争议数据】：联网中心返回记账包中争议数据，数据库中标记为争议数据的总金额及笔数。【快照】

后期增加“数据查询”功能后，点击应跳转至争议数据列表

#### 数据包实时状态



\*数据更新方式：每10分钟自动更新

【打包数量】：当日实时产生的原始交易消息包

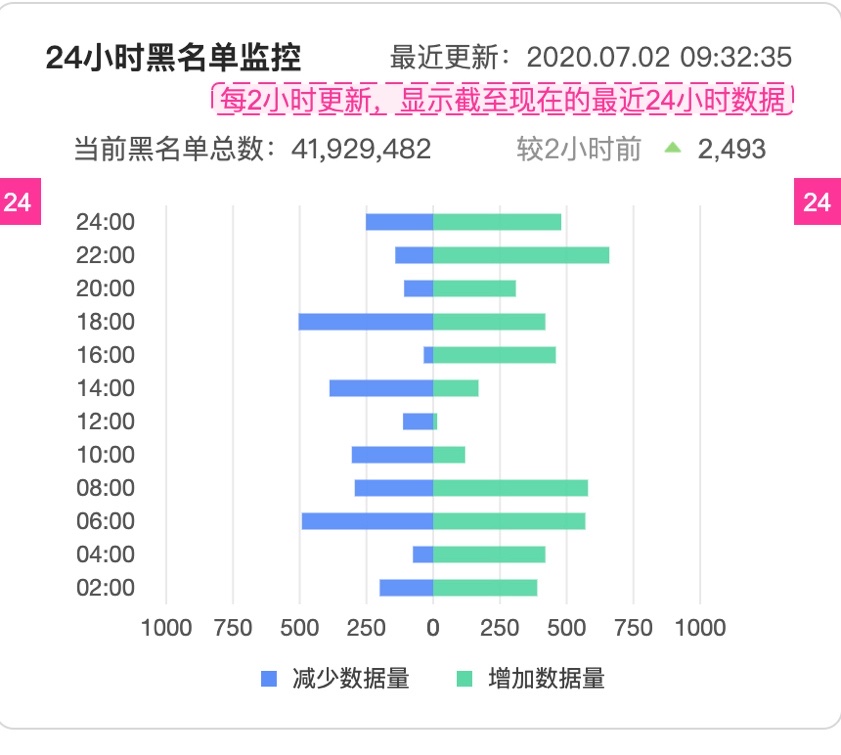
【发送包数量】：当日向联网中心发送消息包数量

【记账包数量】：当日联网中心返回的记账包数量

【应答包数量】：当日联网中心返回的应答包数量

【可视化图表】：每10分钟向数据库中取数据做快照，图表按照快照数据做展示，将产生数据包数量、打包数量、发送包数量、记账包数量、分别以曲线图展示，支持选择时间分段查看对应时间段的数据

#### 黑名单数据量监控

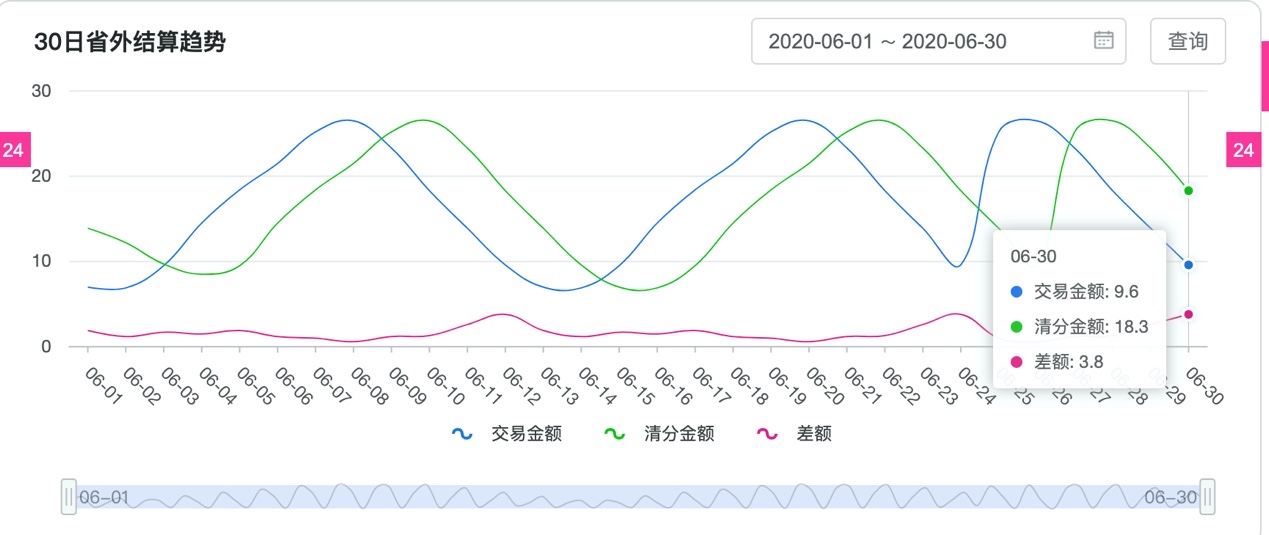


\*数据更新方式：每小时自动更新

【可视化图表】每2小时从数据库中获取省内、省外黑名单数据量，确保每小时黑名单数据更新正常

#### 30日省外结算趋势

\*数据更新方式：每天重新统计更新

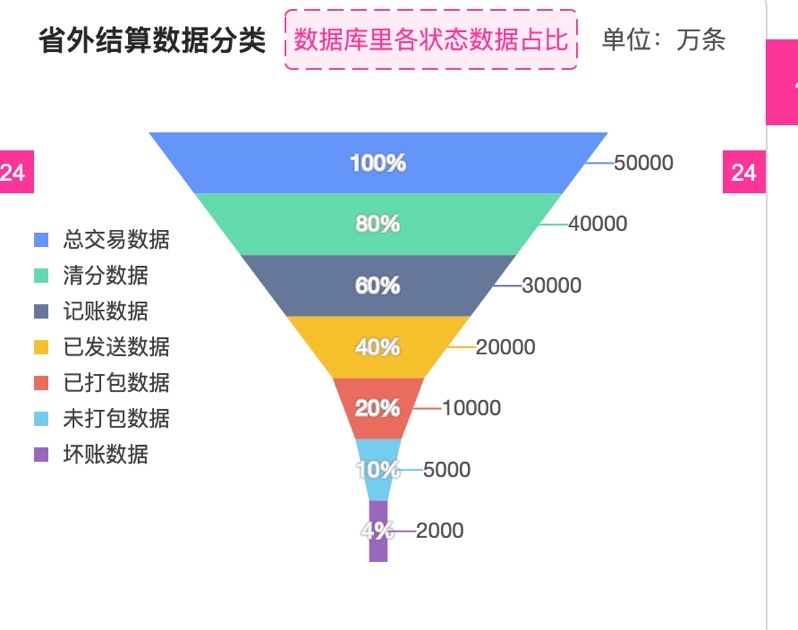


【时间搜索】：默认为截至当前日期前一天的最近30天，支持重新选择时间，范围最大为30天，最少为1天，点击查询后，下方图表根据查询结果更新

【可视化图表】：默认展示到前一天的最近30天省外结算数据。展示交易金额、清分金额、差额

#### 省外结算数据占比

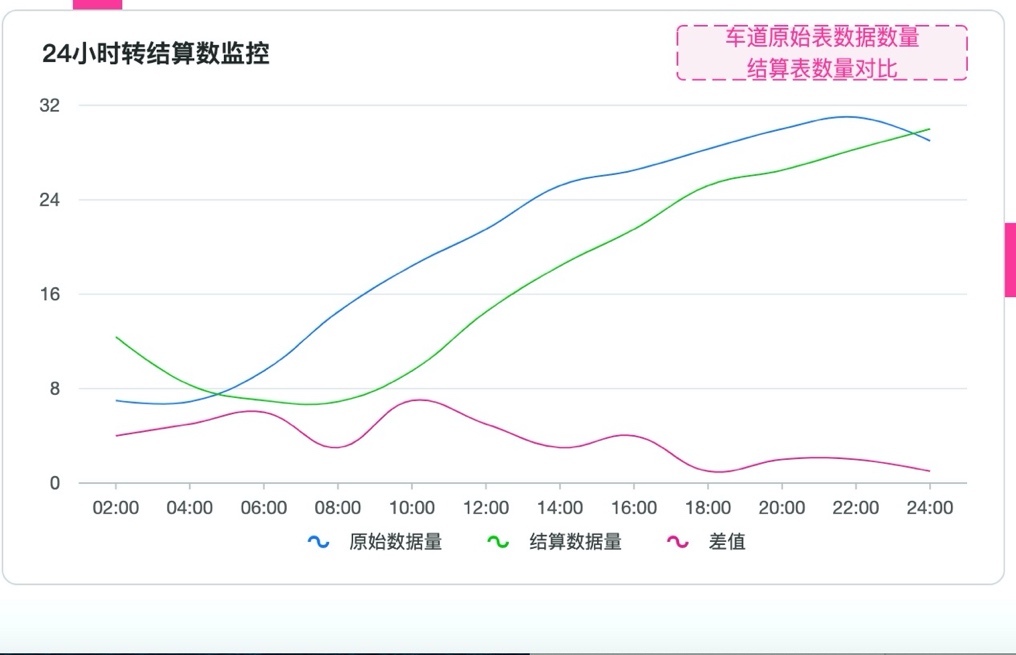
\*数据更新方式：每次刷新页面读取库中数据占比信息【是数据库存取还是请求时计算？？？】



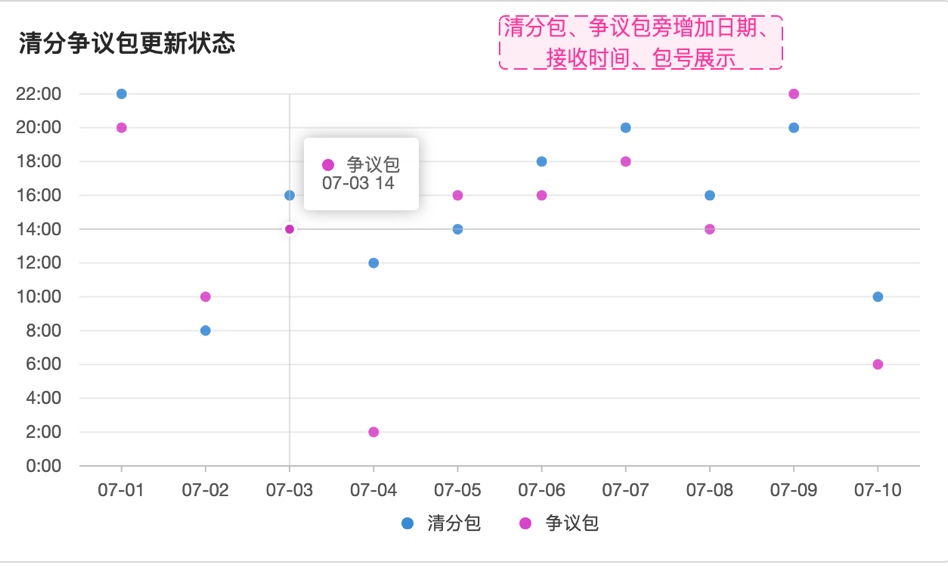
目前数据库中总的省外交易数据量

根据结算状态按比例展示，包括清分、记账、已发送、已打包（未发送）、未打包、争议、坏账数量等

#### 24小时转结算数监控



#### 清分包、争议包接收状态展示



# 补充材料

## 在线版原型地址

<https://axhub.im/ax9/d92fdb9943c5c328>

# 技术概述

Golang web框架：gin；

数据库：mysql；

使用Gorm操作mysql数据库

# 接口说明

## 省外结算平台数据监控

### 累计省外结算数据表的总条数、总金额【被轮巡】

1. 每天定时查询一次结算数据表【b\_js\_jiessj】中产生的省外的结算数据 ，统计总金额、总条数、统计时间；
2. 把统计结果（总金额、总条数、统计时间）插入到结算统计表（新建表【b\_jsjk\_jiesattj】）；

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 备注 | |
| f\_nb\_zje | Int64 | 总金额（分） | |
| F\_Nb\_zts | Int | 总条数 |
| FDtkstjsj | Datetime | 开始统计时间  2020-5-20 13:14:20 |
| FDttjwcsj | Datetime | 统计完成时间  2020-5-20 13:14:20 |
| FNbKawlh | int | 卡网络号类型 |
| F\_NB\_Id |  | 自增id |
| FDtShengntjwcsj | Datetime | 统计日期 |
|  |  | 停车场id |

1. 查询统计结果
2. 获取实时数据，把实时数据累加到统计结果中，把结果存缓存返回给前端

#### 接口设计

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
|  |  | N |  |
|  |  | N |  |

返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| Amount | String | Y | 省外总金额（元） |
| Count | Int | Y | 省外总条数 |

### 查询数据库中省外已清分的交易 总条数、总金额

1）每天定时查询一次结算数据表中【 b\_js\_jiessj 】省外清分数据，统计清分总金额、清分总条数、统计时间；

2）把统计结果（总金额、总条数、统计时间）插入到省外结算清分统计表（新建表【b\_swj\_qingftj】）；

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 备注 |
| Amount | Int64 | 总金额（分） |
| Count | Int | 总条数 |
| Datetime | 2020-5-20 13:14:20 | 统计时间 |
| Id |  | 自增id |

3）查询统计结果

4）把统计结果返回给请求端

#### 接口设计

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
|  |  | Y | 查询日期 |

返回参数：返回最新一条数据

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| Amount | String | Y | 省外已清分总金额 |
| Count | Int | Y | 省外已清分总条数 |

### 查询省外结算数据中存在争议的数据总条数、总金额

1）每天定时查询结算数据表【b\_js\_jiessj】中争议数据的条数、金额，统计时间；

1. 把统计结果（总金额、总条数、统计时间）插入到省外结算争议统计表（新建表【 b\_swjs\_zhengytj 】）；

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 数据类型 | 备注 |  |
| FNbZongje | Int64 | 总金额（分） | Amount |
| FNbZongts | Int | 总条数 | Count |
| FDtTongjsj | 2020-5-20 13:14:20 | 统计时间 | Datetime |
| FNbKawlh | Int | 卡网络号 | CardNetwork number |
| Id |  | 自增id |  |

3）查询统计结果

4）把统计结果返回给请求端

#### 接口设计

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
|  |  |  | 默认当天 |
|  |  |  |  |

返回参数：返回最新争议数据

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| Amount | Int64 | Y | 总金额（分） |
| Count | Int | Y | 总条数 |

### 查询待处理的异常数据 总条数、总金额

1）定时查询一次异常数据表

【b\_dd\_chedckyssjlycb + b\_zdz\_chedckyssjlycb】，

2）把统计结果（总金额、总条数、统计时间）插入到结算待处理异常数据统计表（新建表【b\_swjs\_yicsjtj】）；

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 备注 |
| Id | Int64 | 唯一自增 |
| ZongdzycCount | Int | 总条数（总对总） |
| ZongdzycAmount | Int64 | 总金额（分） |
| DdycCount | Int | 总条数（单点） |
| DdycAmount | Int64 | 总金额（分） |
| Datetime |  | 开始统计时间 单点） |
| Datetime |  | 统计完成时间单点） |
| Datetime |  | 开始统计时间 总对总） |
| Datetime |  | 统计完成时间总对总） |

3）查询统计结果

4）返回结果给前端

#### 接口设计

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
|  |  |  |  |

返回参数：返回最新的

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| Amount | Int64 | Y | 待处理的异常数据的总金额 |
| Count | Int64 | Y | 待处理的异常数据的总条数 |

### 数据包实时状态监控

1）定时10 分钟 扫表（原始交易消息表）【b\_js\_yuansjyxx 】F\_DT\_DABSJ查询出今日的打包数量【F\_NB\_FASZT=0】、今日已发送原始交易消息包数量【F\_NB\_FASZT=2】、原始交易消息应答包数量【b\_js\_yuansjyydxx】F\_DT\_JIESSJ、记账包数量【b\_js\_jizclxx】 F\_DT\_JIESSJ ，

2）把查询结果记录到结算数据包监控表【b\_jsjk\_jiessjbtj】（新建表）

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 备注描述 |  |
| FNbDabsl | Int | 今日的打包数量 | packageCount |
| FNbFassl | int | 已发送原始交易消息包数量 | SendpackegeCount |
| FNbYuansjyxxydsl | int | 原始交易消息应答包数量 | ReplyPackageCount |
| FNbJizbsl | Int | 记账包 | keepAccountPackage  Count |
| FDtKaistjsj |  | 开始统计时间 | Datetime |
| FDtTongjwcsj |  | 统计完成时间（统计时间时间差） |  |
| FNbDabje | Int | 今日的打包金额 | package Amount |
| FNbFasje | int | 已发送原始交易消息包金额 | Sendpackege Amount |
| FNbYuansjyxxydbh | int | 原始交易消息应答包包号 | ReplyPackageBianhao |
| FNbJizbje | Int | 记账包金额 | keepAccountPackage Amount |
| Id | Int64 | 唯一自增 |  |
|  |  | 快照时间 |  |
|  |  |  |  |

3）取最新两个时刻的记录做差值【或者直接按时间段查询】

4）把差值结果返回给请求端

5）可以把某一个时间段的所有取值返回给请求端

#### 接口设计

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
|  |  |  | 条数 |
|  |  |  |  |
|  |  |  |  |

返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| packageCount | Int | Y | 打包数量 |
| SendpackegeCount | Int | Y | 发送包数量 |
| ReplyPackageCount | Int | Y | 应答包数量 |
| keepAccountPackageCount | Int | Y | 记账包数量 |

### 前30天省外结算趋势

1）每天定时查询1次结算数据表【 b\_js\_jiessj 】，获取交易金额、清分金额；

2）计算差额，把交易金额、清分金额差额、统计时间，插入省外结算趋势表【b\_jsjk\_Shengwjsqs】（新建表）

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 备注描述 |
| Id | Int64 | 唯一自增 |
| TransactionAmount | Int64 | 交易金额 |
| clearingAmount | Int64 | 清分金额 |
| diffAmount | Int64 | 计算差额 |
| TransactionCount | Int64 | 交易 条数 |
| clearingCount | Int64 | 清分 条数 |
| diffCount | Int64 | 计算 条数 |
| Datetime |  | 开始统计时间 |
| Datetime |  | 统计完成时间 |
|  |  | 日期 |

3）把数据返回给请求端

#### 接口设计

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| TransactionAmount | Int64 | Y | 交易金额 |
| clearingAmount | Int64 | Y | 清分金额 |
| diffAmount | Int64 | Y | 计算差额 |

返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| TransactionAmount | Int64 | Y | 交易金额 |
| clearingAmount | Int64 | Y | 清分金额 |
| diffAmount | Int64 | Y | 计算差额 |

### 全天24小时黑名单监控？？？

1）每2个小时查询一下黑名单总数

2）把黑明单总数存入黑名单监控表（新建），插入：黑名单总数、更新时间、

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据类型 | 备注 |
| Id | Int64 | 唯一自增 |
| Blacklist | Int | 黑名单总数 |
| AddCount | Int | 数据增加量 |
| decreaseCount | Int | 数据减少量 |
| Datetime |  | 开始统计时间 |
| Datetime |  | 统计完成时间 |
|  |  | 快照时间 |

3）查询黑名单监控表最新2条记录，算出较2小时前数据变化量

4）返回最新黑名单总数、更新时间、数据增加量、数据减少量

#### 接口设计

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
|  |  |  |  |

返回参数：固定12条数据

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| Blacklist | Int | Y | 黑名单总数 |
| AddCount | Int | Y | 数据增加量 |
| decreaseCount | Int | Y | 数据减少量 |
|  |  |  |  |

### 省外结算数据分类

1）数据库里各状态数据占比

2）每天查询一次结算表【 b\_js\_jiessj 】

3）获取数据，处理数据【是否建表存储数据库各状态占比数据？？】

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 备注描述 |
| Id | Int64 | 唯一自增 |
| SwjyCount | In t | 总交易条数 |
| clearingAmount | Int | 清分数据 |
| keepAccountCount | Int | 记账数据 |
| disputeCount | Int | 待处理争议数据 |
| unpackedCount | Int | 未打包数据 |
| packedCount | Int | 已打包数据 |
| SendCount | Int | 已发送数据 |
| BadDebtsCount | Int | 坏账数据 |
| Datetime |  | 开始统计时间 |
| Datetime |  | 统计完成时间 |

4）返回（省外）总交易条数、清分数据、记账数据、待处理争议数据、未打包数据、已打包数据、已发送数据、坏账数据、统计时间等

#### 接口设计

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
|  |  |  |  |

返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| SwjyCount | In t | Y | 总交易条数 |
| clearingAmount | Int | Y | 清分数据 |
| keepAccountCount | Int | Y | 记账数据 |
| disputeCount | Int | Y | 待处理争议数据 |
| unpackedCount | Int | Y | 未打包数据 |
| packedCount | Int | Y | 已打包数据 |
| SendCount | Int | Y | 已发送数据 |
| BadDebtsCount | Int | Y | 坏账数据 |
|  |  |  |  |

### 全天24小时转结算数监控【展示方式？？？】

1）定时查询车道原始包数据数据量b\_dd\_chedckyssj，b\_zdz\_chedckyssj

2）查询结算表数据量【 b\_js\_jiessj 】

3）计算差值

4）把车道原始包数据数据量、结算表数据量、计算差值 存储到转结算数据监控表【b\_jsjk\_zhuanjssjjk】（新表）

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 备注 |
| Id | Int64 | 唯一自增 |
| YuansCount | Int | 车道原始包数据数据量（单点） |
| JiesCount | Int | 结算表数据量 |
| diffCount | Int | 差值 |
| Datetime |  | 开始统计时间 |
| Datetime |  | 统计完成时间 |
| YuansCount | Int | 车道原始包数据数据量（zdz） |
| Datetime |  | 开始统计时间（zdz） |
| Datetime |  | 统计完成时间（zdz） |

1. 查询结算数据监控表，把结果返回给请求端

#### 接口设计

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| YuansCount | Int | Y | 车道原始包数据数据量 |
| JiesCount | Int | Y | 结算表数据量 |
| diffCount | Int | Y | 差值 |
|  |  |  |  |

返回参数：List

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| YuansCount | Int | Y | 车道原始包数据数据量 |
| JiesCount | Int | Y | 结算表数据量 |
| diffCount | Int | Y | 差值 |
|  |  |  | 时间刻度 |

### 清分、争议包更新状态监控

便于定位清分争议包接收的时间以及是否接收失败

1）每天定时查询【b\_js\_qingftjxx】表、【b\_js\_zhengyjyclxx】表

2）查询清分包号【F\_NB\_XIAOXXH】、争议包的包号【F\_NB\_XIAOXXH】、接收时间【F\_DT\_JIESSJ】、统计时间

3）返回数据给前端

#### 接口设计

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| Qingfbh | Int64 | Y | 清分包号 |
| Datetime |  | Y | 清分包接收时间 |
| Zhengybh | Int64 | Y | 争议包的包号 |
| Datetime |  | Y | 争议包接收时间 |

返回参数：List

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| Qingfbh | Int64 | Y | 清分包号 |
| Datetime |  | Y | 清分包接收时间 |
| Zhengybh | Int64 | Y | 争议包的包号 |
| Datetime |  | Y | 争议包接收时间 |
|  |  |  | 刻度日期 |

### 清分核对

在原表中增加字段：昨天要清分金额、是否存在争议、清分标记

## 省内结算平台数据监控

### 累计省内结算数据表的总条数、总金额【实时】

1）每天定时查询结算数据表【 b\_js\_jiessj 】中产生的省内的结算数据 ，统计总金额、总条数、统计时间；

2）把统计结果（总金额、总条数、统计时间）插入到结算统计表（新建表）；

3）获取实时数据，把实时数据插入省内结算统计表

4）实时的把数据查询出来，取省外结算统计表最新的数据，返回给请求端；

#### 接口设计

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| amount | Int64 | Y | 总金额（分） |
| count | Int | Y | 总条数 |

返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| amount | Int64 | Y | 总金额（分） |
| count | Int | Y | 总条数 |

### 查询省内的已发送 总条数、总金额【实时】

1）每天定时查询结算数据表中省内已发送数据，统计总金额、总条数、统计时间；

2） 把查询结果插入到省内的已发送数据统计表（新建）

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 备注 |
| Id | Int | 唯一自增 |
| amount | Int64 | 总金额（分） |
| count | Int | 总条数 |
| Datetime |  | 开始统计时间 |
| Datetime |  | 统计完成时间 |

3）把统计结果返回给请求端

#### 接口设计

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| amount | Int64 | Y | 总金额（分） |
| count | Int | Y | 总条数 |

返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| amount | Int64 | Y | 总金额（分） |
| count | Int | Y | 总条数 |

### 查询省内已请款的数据总条数、总金额

1）每天定时查询结算数据表【 b\_js\_jiessj 】中已请款数据的条数、金额

2）把查询的数据插入到省内请款统计表（新建）

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 备注 |
| Id | Int | 唯一自增 |
| amount | Int64 | 总金额（分） |
| count | Int | 总条数 |
| Datetime |  | 开始统计时间 |
| Datetime |  | 统计完成时间 |

3）把统计结果返回给请求端

#### 接口设计

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| amount | Int64 | Y | 总金额（分） |
| count | Int | Y | 总条数 |

返回参数：返回最新的数据

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| amount | Int64 | Y | 总金额（分） |
| count | Int | Y | 总条数 |

### 查询坏账（拒付）数据 总条数、总金额

1）定时查询 【b\_js\_jiessj】拒付坏账数据

2）把查询的数据插入到省内坏账（拒付）数据统计表（新建）

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 备注 |
| Id | Int | 唯一自增 |
| amount | Int64 | 总金额（分） |
| count | Int | 总条数 |
| Datetime |  | 开始统计时间 |
| Datetime |  | 统计完成时间 |

1. 返回拒付坏账数据金额、笔数

#### 接口设计

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| amount | Int64 | Y | 总金额（分） |
| count | Int | Y | 总条数 |

返回参数：返回最新一条数据

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| amount | Int64 | Y | 总金额（分） |
| count | Int | Y | 总条数 |

### 省内实时数据

1）每10分钟更新

2）每日定时查询结算表，省内产生数据金额、省内已发送数据金额、省内已记账数据金额

3）把数据插入 省内实时数据监控表（新建）

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 备注描述 |
| Id | Int64 | 唯一自增 |
| amount | Int64 | 产生金额 |
| SendAmount | Int64 | 省内已发送数据金额 |
| keepAccountAmount | int | 省内已记账数据金额 |
| Datetime |  | 开始统计时间 |
| Datetime |  | 统计完成时间 |
| count | Int64 | 产生条数 |
| Send count | Int64 | 省内已发送数据条数 |
| keepAccountcount | int | 省内已记账数据条数 |

4）接收实时数据：产生数据金额、已发送数据金额、已记账数据金额

5）接收的实时数据处理（累加到缓存中）

6）返回数据给前端

#### 接口设计

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| amount | Int64 | Y | 产生金额 |
| SendAmount | Int64 | Y | 省内已发送数据金额 |
| keepAccountAmount | int | Y | 省内已记账数据金额 |
|  |  |  |  |

返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| amount | Int64 | Y | 产生金额 |
| SendAmount | Int64 | Y | 省内已发送数据金额 |
| keepAccountAmount | int | Y | 省内已记账数据金额 |
|  |  |  |  |

### 前30日省内结算概览

1）每天定时查询1次结算数据表【 b\_js\_jiessj 】，获取省内交易金额、省内请款金额；

2）计算差额，把交易金额、清分金额差额、统计时间，插入省内结算趋势表（新建表）

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 备注描述 |
| Id | Int64 | 唯一自增 |
| TransactionAmount | Int64 | 交易金额 |
| PleaseAmount | Int64 | 请款金额 |
| diffAmount | Int64 | 计算差额 |
| Datetime |  | 开始统计时间 |
| Datetime |  | 统计完成时间 |

1. 把数据返回给请求端

#### 接口设计

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| TransactionAmount | Int64 | Y | 交易金额 |
| PleaseAmount | Int64 | Y | 请款金额 |
| diffAmount | Int64 | Y | 计算差额 |
|  |  |  |  |

返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| TransactionAmount | Int64 | Y | 交易金额 |
| PleaseAmount | Int64 | Y | 请款金额 |
| diffAmount | Int64 | Y | 计算差额 |
|  |  |  |  |

### 海岭数据同步监控

1）获取海玲结算数据总量、已同步的数据量

2）把数据插入数据同步监控表（新建）

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 备注 |
| Id | Int64 | 唯一自增 |
| Count | Int64 | 海玲结算数据总量 |
| syncCount | Int | 已同步的数据量 |
| Datetime |  | 开始统计时间 |
| Datetime |  | 统计完成时间 |

3）返回数据给前端

#### 接口设计

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| Count | Int64 | Y | 海玲结算数据总量 |
| syncCount | Int | Y | 已同步的数据量 |

返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| Count | Int64 | Y | 海玲结算数据总量 |
| syncCount | Int | Y | 已同步的数据量 |
|  |  |  | 时间刻度 |

### 省内结算数据分类

1）每天查询一次结算表【 b\_js\_jiessj 】

2）查询结算表里：省内总交易数据量、请款数据、未发送数据、发送数据、拒付数据

3）获取数据，处理数据

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 备注描述 |
| Id | Int64 | 唯一自增 |
| ShengnAmount | Int | 省内总交易数据量 |
| PleaseAmount | Int | 请款数据 |
| UnsendCount | Int | 未发送数据 |
| SendCount | Int | 发送数据 |
|  |  |  |
| RefuseCount | Int | 拒付数据 |
| Datetime |  | 开始统计时间 |
| Datetime |  | 统计完成时间 |

4）省内总交易数据量、请款数据、未发送数据、发送数据、拒付数据、统计时间等

5）把数据返回给前端

#### 接口设计

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| ShengnAmount | Int | Y | 省内总交易数据量 |
| PleaseAmount | Int | Y | 请款数据 |
| UnsendCount | Int | Y | 未发送数据 |
| SendCount | Int | Y | 发送数据 |
|  | Int | Y | 未打包数据 |
| RefuseCount | Int | Y | 拒付数据 |

返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| ShengnAmount | Int | Y | 省内总交易数据量 |
| PleaseAmount | Int | Y | 请款数据 |
| UnsendCount | Int | Y | 未发送数据 |
| SendCount | Int | Y | 发送数据 |
|  | Int | Y | 未打包数据 |
| RefuseCount | Int | Y | 拒付数据 |

### 异常数据停车场top10

1）累计未处理异常数据数量最多的停车场 快照

2）查询【b\_dd\_chedckyssjlycb 、 b\_zdz\_chedckyssjlycb】异常数据中出现最多的停车场

3）返回数据给前端：List

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 备注 |
| Id | Int64 | 唯一自增 |
| ZongdzycCount | Int | 总条数（总对总） |
| ZongdzycAmount | Int64 | 总金额（分） |
| DdycCount | Int | 总条数（单点） |
| DdycAmount | Int64 | 总金额（分） |
| Gongsid | String | 停车场名称 |
| Datetime |  | 开始统计时间 |
| Datetime |  | 统计完成时间 |
|  |  | 统计日期 |

#### 接口设计

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| Gongsid | String | Y | 停车场名称 |
| Count | Int | Y | 异常数据数量 |

返回参数：List

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| Gongsid | String | Y | 停车场名称 |
| Count | Int | Y | 异常数据数量 |

### 逾期数据停车场top10

1）提交逾期数据数量最多的停车场，默认展示最近30天

2）查询最近30天逾期数据中出现最多的停车场

3）统计【 b\_zdz\_chedckyssjlycb】表中逾期数据【F\_VC\_YICLX =21】

返回数据给前端

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 备注 |
| Id | Int64 | 唯一自增 |
| Count | Int | 逾期总条数 |
| Amount | Int64 | 逾期总金额（分） |
| Gongsid | String | 停车场名称 |
| Datetime |  | 开始统计时间 |
| Datetime |  | 统计完成时间 |

#### 接口设计

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
|  |  |  | 默认 |

返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 描述 |
| Gongsid | String | Y | 停车场名称 |
| Count | Int | Y | 逾期数据数量 |
| Datetime |  | Y | 统计时间 |