**联网结算模块实现设计**

# 一、目标

**阶段1目标**

实现从海岭Oracle数据库向联网发送数据:

**阶段2目标**

实现从Mysql数据库向联网发送数据；数据库字段表一致。

# 二、功能实现

## 2.1 参照海岭现有代码实现功能

开4个线程:

线程1： 从车道消费数据表中一次查找100条未打包的数据，(5分钟或100条)，按照联网交互的格式要求生成XML并按照7z方式打包；存入指定的文件夹中；并更新数据库打包标记；注意使用事务更新包号，打包时间。

线程2：从打包文件夹中定期扫描文件，若有则读取，按照联网规范发送至联网的数据端口中；发送成功后将文件转移至另外文件夹；

线程3：服务监听端口，接收联网发送过来的数据包，并存储在指定文件夹；

线程4：定期扫描接收联网的接收数据的文件夹，解压后分析包。更新应答包状态、争议包状态、记账包状态、清分包状态、退费包状态；

## 2.2 数据核对结果

1、每日发送的数据包，哪些没有回记账包；没有回的在运维过程中重发功能；

2、清分包的数据总数是否与发送的数据一致，数据库记录差异；

## 2.3 异常处理及日志

更新数据库异常、打包异常、发送异常的错误处理机制及日志记录；

# 三、其他

## 3.1 文档完善

将更新的字段及内容含义在文档中更新，设计文档更新。

## 3.2 时间要求

本月底之前完成开发及测试；

# 四、接口规范

**1、数据库**：mysql数据库；用golang xorm操作mysql数据库；  
  
**2、线程1：**   
 从原始交易结算数据表中一次查找100条未打包的记录数据，(5分钟或100条)，未打包数据的数据有标记；  
 按照联网交互的格式要求生成XML文件存于generatexml文件夹。 并按照 LZ77 方式打包；存入指定的文件夹中sendzipxml；

并更新数据库打包标记；注意使用事务更新包号，打包时间。

具体步骤：

查询原始交易数据（在数据层）

准备数据（在数据层）

Xml数据生成Xml文件、压缩，存文件

Md5计算

7z压缩（Cgo解决）

插入原始交易消息包记录

更新原始交易数据的状态

Xml数据格式校准：

一个原始交易消息包type Message struct {  
 XMLName xml.Name `xml:"Message"`  
 Header Header `xml:"Header"`  
 Body Body `xml:"Body"`  
}  
  
type Header struct {  
 XMLName xml.Name `xml:"Header"`  
 Version string  *//统一 00010000 Hex(8) Header* MessageClass int32  *//消息传输的机制* MessageType int32  *//消息的应用类型* SenderId string  *// Hex(16位，不足补零) 发送方Id* ReceiverId string  *//Hex(16位，不足补零) 接收方Id* MessageId int64  *//消息序号，从1开始，逐1递增 ，8字节*}  
  
type Body struct {  
 XMLName xml.Name `xml:"Body"`  
 ContentType int `xml:",attr"` *//始终为1* ClearTargetDate time.Time *//日期 如：2017-06-05 清分目标日期* ServiceProviderId string *//通行宝中心系统Id，表示消息包中的交易是由收费方产生的* IssuerId string *//发行服务机构Id， 表示产生交易记录的发行服务机构。* MessageId int64 *//交易消息包Id。* Count int *//本消息包含的记录数量* Amount int *//交易总金额(分)* Transaction []Transaction *//交易原始数据 `xml:"Message>Body>Transaction"`*}  
type Transaction struct {  
 XMLName xml.Name `xml:"Transaction"`  
 TransId int  *// 包内顺序Id，从1开始递增 ，包内唯一的交易记录* Time time.Time  *//交易的发生时间，需要加TAC计算* Fee int  *//交易的发生金额(分)* Service Service  *//服务信息* ICCard ICCard  *//IC卡信息* Validation Validation  *//与校验相关的信息* OBU OBU  *//参加交易的电子标签信息* CustomizedData string  *//特定发行方与通行宝收费方之间 约定格式的交易信息【 】*}  
*//服务信息*type Service struct {  
 XMLName xml.Name `xml:"Service"`  
 ServiceType int *//交易的服务类型【】* Description string *//对交易的文字解释【数据库中账单描述】* Detail string *//交易详细信息*}  
*//IC卡信息*type ICCard struct {  
 XMLName xml.Name `xml:"ICCard"`  
 CardType int *//卡类型，22为储值卡；23记账卡* NetNo string *//网络编码，BCD码 Hex(4) k网络号（16进制）数据库10进制* CardId string *//IC卡物理编号，BCD码 Hex(16) 卡号* License string *//0015文件中记录的车牌号* PreBalance string *//交易前余额，以元为单位 Decimal* PostBalance string *//交易后余额，以元为单位 Decimal*}  
  
*//主要用于TAC计算*type Validation struct {  
 XMLName xml.Name `xml:"Validation"`  
 TAC string *//交易时产生的TAC码，8位16进制数 Hex(8)* TransType string *//交易标识，2位16进制数，PBOC定义，如06为传统交易，09为复合交易 Hex(2)* TerminalNo string *//12位16进制数据，即PSAM号，PSAM中0016文件中的终端机编号 Hex(2)* TerminalTransNo string *//8位16进制数，PSAM卡脱机交易序号，在MAC1计算过程中得到 Hex(8)*}  
  
type OBU struct {  
 XMLName xml.Name `xml:"OBU"`  
 NetNo string *//4501 OBU网络号* OBUId string *//OBU物理编号，BCD码 4501191509252866* OBEState string *//2字节的OBU状态* License string *//OBU中记录的车牌号*}

**3、线程2**：  
 从文件夹sendzipxml中 定期 扫描打包文件，  
 若有则读取文件，并按照联网规范 发送至 联网中心的数据服务端口中；  
 发送成功后（收到联网中心的ok 应答） 将文件转移至另外文件夹sendsucceed；

具体步骤：

定时器定期扫描sendzipxml文件

读取文件

准备报文

发送报文

发送成功则mv消息包至sendsucceed

发送失败 触发重发机制（）

tcp通讯报文格式：

type SendStru struct {  
 Massageid [20]byte *//消息报文序号 20字节Char型，不足左补0* Xml\_length [6]byte *//压缩后XML消息包长度* Md5\_str [32]byte *//32字节MD5校验串 MD5值由<Message>* Xml\_msg []byte *//二进制压缩后的XML 消息包 .xml.lz77*}  
  
tcp通讯报文应答格式：type ReplyStru struct {  
 Massageid [20]byte *//消息报文序号【20字节】* result byte *//【1字节】 1 成功接收 0 接收超时、长度不符*}

**4、线程3**：  
 服务端监听联网中心的端口，接收联网发送过来的数据包，并存储在指定文件夹receive；  
【 服务端监听端口，接收各种数据包，把数据写成文件，存放入指定文件夹 receive 】

具体步骤：

监听联网中心端口

接收数据包

存储数据包至receive文件夹  
  
**5、线程4**：  
 定期扫描 接收联网的接收数据的文件夹 receive，如果有文件就解压， 解压后分析包。  
 1、更新包状态，更新应答包状态、争议包状态、记账包状态、清分包状态、退费包状态；

2、处理包信息【】

具体步骤：

定时器定期扫描receive文件夹

读取文件

解析文件

应答数据包

记账数据包

争议数据包

清分数据包

退费数据包

处理文件【待解决】

# 五、测试用例