Обработка 1С8.3 (толстый/тонкий клиент, управляемая форма)

Конфигурация BP3

**п1.**

Обработкой производится запрос на сервис

<http://localhost:8033/LoadDocs>

**п2.**

В ответ обработка получает

массив массивов полей (описанных ниже) в формате json

Поля массива

// 0 id: Integer; ИД строки

// 1 session\_id: Integer; ИД смены

// 2 sessionnum: Integer; Номер смены

// 3 azscode: String; Код АЗС

// 4 azsextcode: String; Код АЗС в 1С

// 5 dir: string; Направление операции («Расход», «Приход»)

// 6 sdate: TDateTime; Дата смены

// 7 clientcode: String; Код контрагента

// 8 clientname: String; Наименование

// 9 paymentcode: String; Код вида платежа (разные, для перелива 99PL99 – «безналичный платёж»

//10 paymentname: String; Наименование вида платежа

//11 fuelcode: String; Код товара (ГСМ)

//12 fuelname: String; Наименование товара

//13 volume: Extended; Объём

//14 price: Extended; Цена

//15 density: Extended; Плотность

//16 mass: Extended;

//17 nds: String;

//18 sumnds: Extended; Сумма НДС

//19 whole: Extended; сумма С НДС

//20 amount0: Extended; сумма Без НДС

//21 номер договора

//22 склад получатель (наименование)

Фактически, имеем таблицу документов (далее «входной массив»)

**Задача**

**п3.**

Сгруппировать по АЗС, Складу, Контрагенту, Виду платежа, Товару, Цене

Суммировать Массу, Объём, Суммы.

Прочие поля не учитываем при суммировании

**п4.**

Результирующую таблицу (далее «Документы») вывести на форму и дать возможность пользователю редактировать поля

**п5.**

Сформировать таблицу плотностей по входному массиву

|  |  |
| --- | --- |
| Товар | Плотность |

Без дублирующихся строк

Вывести таблицу (далее «Плотности») на форму с возможностью редактирования плотности.

**п6.**

Сформировать и записать документы РеализацияТоваровУслуг («Расход»), Или Перемещение, приход, что там в конфе БП3 положено

(опционально - записать и провести)

Для вида оплаты PL99PL выбирать контрагента указанного пользователе как «Контрагент ПЛ» (см. картинку №1)

**п7.**

Учитывать заданный пользователем интервал дат, АЗС, склад и список контрагентов.

Плотность для документа берётся из таблицы «Плотности» усреднённая за указанный интервал дат по товару.

**п8. (самое хитрое)**

Сформировать ответ сервису.

По каждой (!) строке входного массива:

Если строка входит в успешно записанный документ сформировать ответ «**ИДстроки#ИДСмены#Y**”

Если строка не учтена (не просуммирована) в записанном документе, сформировать ответ «**ИДстроки#ИДСмены#N##Причина отказа или ошибка**”

Причины отказа: не входит в интервал дат, в список контрагентов, не соответствует склад или АЗС (см. п7)

Ошибки: Ошибки записи (проводки), самые разные. Например, не найден товар, не найден контрагент, то есть то, что вернёт процедура записи

try

doc.Записать(post);

except

Инфо = ИнформацияОбОшибке();

ош = Инфо.Описание;

Сообщить("Описание='" + ош + "'");

success = "N";

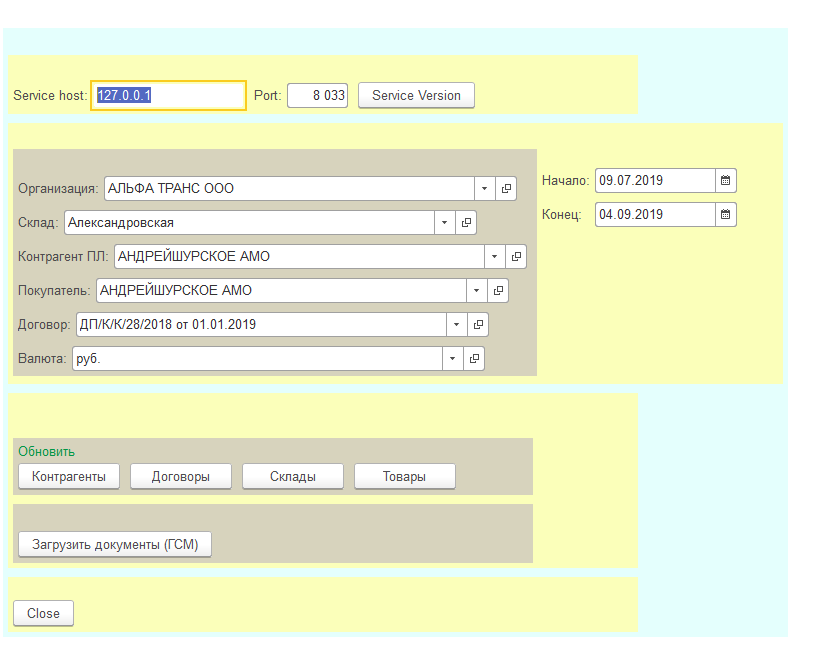
endtry;

**п9.**

Полученные строки объединить в одну строкой «@@» и отправить POST запрос на сервис http://localhost:8033/PostLDResults

**п10.**

Картинка №1



Пояснения:

Вместо поля «покупатель» должен быть выбор контрагента или формирования списка контрагентов (см. п7)

Поле «договор» не нужно. При записи документа брать актуальный договор для контрагента, или из входного массива (тут неясность)

(от заказчика:

*договора выбирать последний по актуальности , или привязанный к контрагенту*

*При создании документов выбирать активный договор для топлива и для товара, или дать возможность изменять или выбирать*)

На форму:

Добавить поле «АЗС» для выбора АЗС по которой формировать документы для записи

Добавить кнопку «записать документы»

Добавить таблицы «Документы» и «Плотности»

**п11.** Приложение