ПРИЛОЖЕНИЕ А

(обязательное)

Листинг программы

Функция ПолучитьСсылкуНаЗначениеПеречисления( пИмяПеречисления, пИмяЗначения ) Экспорт

Попытка

УбратьПробелы = СокрЛП(ТРег(пИмяЗначения));

Возврат Перечисления[пИмяПеречисления][СтрЗаменить(УбратьПробелы, " " , "")];

исключение

Сообщить("Ошибка получения ссылки на значение перечисления. " + пИмяПеречисления + " :: " + УбратьПробелы);

Возврат неопределено;

КонецПопытки;

КонецФункции

Функция Auth(Data)

Если ЗначениеЗаполнено(Data) Тогда

ПолученныеДанные = ОбщегоНазначения.Десериализатор(Data);

КонецЕсли;

Пользователь = Справочники.Пользователи.НайтиПоРеквизиту("Логин", ПолученныеДанные.Логин);

Если Не ПустаяСтрока(Пользователь) Тогда

Если Пользователь.Пароль = ПолученныеДанные.Пароль Тогда

Данные = Новый Структура("ФИО, Роль", Пользователь.Наименование, Строка(Пользователь.Роль));

Возврат ОбщегоНазначения.Сериализовать(Данные);

КонецЕсли;

КонецЕсли;

КонецФункции

Функция MobileToCentral(Data)

Если ЗначениеЗаполнено(Data) Тогда

ПолученныеДанные = ОбщегоНазначения.Десериализатор(Data);

Иначе

Сообщить("Произошла ошибка, данные не получены");

Возврат "Произошла ошибка, данные не получены"

КонецЕсли;

Для каждого Данные Из ПолученныеДанные Цикл

СоздатьДокументОтгрузка = Документы.Отгрузка.СоздатьДокумент();

СоздатьДокументОтгрузка.Дата = Данные.Дата;

СоздатьДокументОтгрузка.Уникальность = Данные.Уникальность;

СоздатьДокументОтгрузка.Автомобиль = Справочники.Автомобили.НайтиПоНаименованию(Данные.Автомобиль);

Для Каждого СтрокаТабличнойЧасти Из Данные.МассивКонтактнойИнформации Цикл

Стр = СоздатьДокументОтгрузка.Товары.Добавить();

Стр.УникальныйКод = СтрокаТабличнойЧасти.УникальныйКод;

Стр.ВидОтправления = Справочники.ВидОтправления.НайтиПоНаименованию(СтрокаТабличнойЧасти.ВидОтправления);

Стр.Длинна = СтрокаТабличнойЧасти.Длинна;

Стр.Ширина = СтрокаТабличнойЧасти.Ширина;

Стр.Высота = СтрокаТабличнойЧасти.Высота;

Стр.Вес = СтрокаТабличнойЧасти.Вес;

Стр.Объем = СтрокаТабличнойЧасти.Объем;

Стр.Стоимость = СтрокаТабличнойЧасти.Стоимость;

КонецЦикла;

КонецЦикла;

СоздатьДокументОтгрузка.Записать(РежимЗаписиДокумента.Проведение);

Возврат "Данные успешно загружены"

КонецФункции

Функция MobileToCentralShipmentStatuses(Data)

Если ЗначениеЗаполнено(Data) Тогда

ПолученныеДанные = ОбщегоНазначения.Десериализатор(Data);

Иначе

Сообщить("Произошла ошибка, данные не получены");

Возврат "Произошла ошибка, данные не получены"

КонецЕсли;

Для каждого Данные Из ПолученныеДанные Цикл

РегСвед = РегистрыСведений.СтатусыОтправлений.СоздатьМенеджерЗаписи();

РегСвед.Ключ = Справочники.КлючАналитикиЗаказов.НайтиПоНаименованию(Данные.Ключ);

РегСвед.Статус = ПолучитьСсылкуНаЗначениеПеречисления("СтатусЗаявки",Данные.Статус);;

РегСвед.Период = Данные.Период;

РегСвед.Склад = Справочники.Склады.НайтиПоНаименованию(Данные.Склад);

РегСвед.Место = Справочники.МестаХранения.НайтиПоНаименованию(Данные.Место);;

РегСвед.Записать(Истина);

КонецЦикла;

Возврат "Данные успешно загружены"

КонецФункции

Функция MobileToCentralTravelOrder(Data)

Если ЗначениеЗаполнено(Data) Тогда

ПолученныеДанные = ОбщегоНазначения.Десериализатор(Data);

Иначе

Сообщить("Произошла ошибка, данные не получены");

Возврат "Произошла ошибка, данные не получены"

КонецЕсли;

Для каждого ДанныеРеквизитовПутевогоЛиста Из ПолученныеДанные Цикл

СоздатьДокументПутевойЛист = Документы.ПутевойЛист.СоздатьДокумент();

СоздатьДокументПутевойЛист.Дата = ДанныеРеквизитовПутевогоЛиста.Дата;

Для Каждого СтрокаТабличнойЧасти Из ДанныеРеквизитовПутевогоЛиста.МассивКонтактнойИнформации Цикл

Стр = СоздатьДокументПутевойЛист.Маршрут.Добавить();

Стр.УидГруза = Справочники.КлючАналитикиЗаказов.НайтиПоНаименованию(СтрокаТабличнойЧасти.УидГруза);

Стр.АдресДоставки = СтрокаТабличнойЧасти.АдресДоставки;

Стр.ВремяПрибытия = СтрокаТабличнойЧасти.ВремяПрибытия;

Стр.ВремяУбытия = СтрокаТабличнойЧасти.ВремяУбытия;

Стр.Доставлено = СтрокаТабличнойЧасти.Доставлено;

КонецЦикла;

СоздатьДокументПутевойЛист.Записать(РежимЗаписиДокумента.Проведение);

КонецЦикла;

Возврат "Данные успешно загружены"

КонецФункции

Функция CentralToMobileAnalyticsKey()

Массив = Новый Массив;

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| КлючАналитикиЗаказов.Наименование КАК Наименование,

| КлючАналитикиЗаказов.ВидОтправления КАК ВидОтправления,

| КлючАналитикиЗаказов.Длинна КАК Длинна,

| КлючАналитикиЗаказов.Ширина КАК Ширина,

| КлючАналитикиЗаказов.Высота КАК Высота,

| КлючАналитикиЗаказов.Вес КАК Вес,

| КлючАналитикиЗаказов.Объем КАК Объем,

| КлючАналитикиЗаказов.Стоимость КАК Стоимость

|ИЗ

| Справочник.КлючАналитикиЗаказов КАК КлючАналитикиЗаказов";

РезультатЗапроса = Запрос.Выполнить();

ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

Структура = Новый Структура;

Структура.Вставить("Наименование", ""+ВыборкаДетальныеЗаписи.Наименование);

Структура.Вставить("ВидОтправления", ""+ВыборкаДетальныеЗаписи.ВидОтправления);

Структура.Вставить("Длинна", ""+ВыборкаДетальныеЗаписи.Длинна);

Структура.Вставить("Ширина", ""+ВыборкаДетальныеЗаписи.Ширина);

Структура.Вставить("Высота", ""+ВыборкаДетальныеЗаписи.Высота);

Структура.Вставить("Вес", ""+ВыборкаДетальныеЗаписи.Вес);

Структура.Вставить("Объем", ""+ВыборкаДетальныеЗаписи.Объем);

Структура.Вставить("Стоимость", ""+ВыборкаДетальныеЗаписи.Стоимость);

Массив.Добавить(Структура);

КонецЦикла;

Возврат ОбщегоНазначения.Сериализовать(Массив);

КонецФункции

Функция CentralToMobileShipmentStatuses()

Массив = Новый Массив;

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| СтатусыОтправлений.Период КАК Период,

| СтатусыОтправлений.Ключ КАК Ключ,

| СтатусыОтправлений.Статус.Ссылка КАК СтатусСсылка,

| СтатусыОтправлений.Склад КАК Склад,

| СтатусыОтправлений.Место КАК Место

|ИЗ

| РегистрСведений.СтатусыОтправлений КАК СтатусыОтправлений";

РезультатЗапроса = Запрос.Выполнить();

ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

Структура = Новый Структура;

Структура.Вставить("Ключ", ""+ВыборкаДетальныеЗаписи.Ключ);

Структура.Вставить("Статус", ""+ВыборкаДетальныеЗаписи.СтатусСсылка);

Структура.Вставить("Склад", ""+ВыборкаДетальныеЗаписи.Склад);

Структура.Вставить("Место", ""+ВыборкаДетальныеЗаписи.Место);

Структура.Вставить("Период", ""+ВыборкаДетальныеЗаписи.Период);

Массив.Добавить(Структура);

КонецЦикла;

Возврат ОбщегоНазначения.Сериализовать(Массив);

КонецФункции

Функция CentralToMobileTravelOrder()

Массив = Новый Массив;

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ПутевойЛист.Дата КАК Дата,

| ПутевойЛист.Маршрут.(

| УидГруза КАК УидГруза,

| АдресДоставки КАК АдресДоставки,

| ВремяПрибытия КАК ВремяПрибытия,

| ВремяУбытия КАК ВремяУбытия

| ) КАК Маршрут

|ИЗ

| Документ.ПутевойЛист КАК ПутевойЛист";

РезультатЗапроса = Запрос.Выполнить();

ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

Структура = Новый Структура;

Структура.Вставить("Дата", ""+ВыборкаДетальныеЗаписи.Дата);

ВыборкаКИ = ВыборкаДетальныеЗаписи.Маршрут.Выбрать();

МассивКИ = Новый Массив;

Пока ВыборкаКИ.Следующий() Цикл

СтруктураКИ = Новый Структура;

СтруктураКИ.Вставить("УидГруза", ВыборкаКИ.УидГруза);

СтруктураКИ.Вставить("АдресДоставки", ВыборкаКИ.АдресДоставки);

СтруктураКИ.Вставить("ВремяПрибытия", ВыборкаКИ.ВремяПрибытия);

СтруктураКИ.Вставить("ВремяУбытия", ВыборкаКИ.ВремяУбытия);

МассивКИ.Добавить(СтруктураКИ);

КонецЦикла;

Структура.Вставить("МассивКонтактнойИнформации", МассивКИ);

Массив.Добавить(Структура);

КонецЦикла;

Возврат ОбщегоНазначения.Сериализовать(Массив);

КонецФункции

Процедура ОбработкаЗаполнения(ДанныеЗаполнения, СтандартнаяОбработка)

Если ТипЗнч(ДанныеЗаполнения) = Тип("ДокументСсылка.СчетПокупателю") Тогда

Договор = ДанныеЗаполнения.Договор;

ЗаявкиКлиентов = ДанныеЗаполнения.ЗаявкиКлиентов;

ОплатаДо = ДанныеЗаполнения.ОплатаДо;

Скидка = ДанныеЗаполнения.Скидка;

СуммаКОплате = ДанныеЗаполнения.СуммаКОплате;

Для Каждого ТекСтрокаТовары Из ДанныеЗаполнения.Товары Цикл

НоваяСтрока = Товары.Добавить();

НоваяСтрока.Вес = ТекСтрокаТовары.Вес;

НоваяСтрока.ВидОтправления = ТекСтрокаТовары.ВидОтправления;

НоваяСтрока.Высота = ТекСтрокаТовары.Высота;

НоваяСтрока.Длинна = ТекСтрокаТовары.Длинна;

НоваяСтрока.Объем = ТекСтрокаТовары.Объем;

НоваяСтрока.Описание = ТекСтрокаТовары.Описание;

НоваяСтрока.Стоимость = ТекСтрокаТовары.Стоимость;

НоваяСтрока.УникальныйКод = ТекСтрокаТовары.УникальныйКод;

НоваяСтрока.Ширина = ТекСтрокаТовары.Ширина;

КонецЦикла;

КонецЕсли;

КонецПроцедуры

Процедура ОбработкаПроведения(Отказ, Режим)

Движения.ДенежныеОбороты.Записывать = Истина;

Движение = Движения.ДенежныеОбороты.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Расход;

Движение.Период = Дата;

Движение.Контрагент = Заказчик;

Движение.Сумма = СуммаКОплате;

    Движения.Записать();

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ДенежныеОборотыОстатки.Контрагент КАК Контрагент,

| ДенежныеОборотыОстатки.СуммаОстаток КАК СуммаОстаток

|ИЗ

| РегистрНакопления.ДенежныеОбороты.Остатки КАК ДенежныеОборотыОстатки

|ГДЕ

| ДенежныеОборотыОстатки.СуммаОстаток < 0";

РезультатЗапроса = Запрос.Выполнить();

ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

Если ВыборкаДетальныеЗаписи.Количество()>0 Тогда

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

Сообщить("Переплата: "+-ВыборкаДетальныеЗаписи.СуммаОстаток);// Вставить обработку выборки ВыборкаДетальныеЗаписи

КонецЦикла;

Отказ=Истина;

КонецЕсли;

КонецПроцедуры