### Объекты и их поля

document:

* hasCard (bool) - флаг наличия карты в чеке: 1 - есть, - нет
* hasActiveCard (bool) - флаг наличия карты лояльности в чеке: 1 - есть, - нет
* basePaymentName (string) - имя базовой валюты в чеке
* identifier (string) - уникальный идентификатор чека
* urlEgais (string) - точный веб-адрес, по которому можно найти акцизную продукцию, присланную в чеке
* digitalSignatureEgais (string) - цифровая подпись акцизной продукции
* registeredPdfInEgais (bool) - флаг того, что акцизная позиция документа была зарегистрирована в ЕГАИС. true - зарегистрирована, false - иначе
* noPdfUrlEgais (string) - точный веб-адрес, по которому можно найти безакцизную продукцию, присланную в чеке
* noPdfDigitalSignatureEgais (string) - цифровая подпись безакцизной продукции
* registeredNoPdfInEgais (bool) - флаг того, что безакцизная позиция документа была зарегистрирована в ЕГАИС. true - зарегистрирована, false - иначе
* hasCoupon (bool) - флаг наличия купона в чеке
* hasCouponGiven (bool) - флаг наличия купона на выдачу в чеке
* shownAlcoholCaution (bool) - флаг факта показа сообщения о предупреждении о скором запрете продажи алкоголя. true - сообщение было показано, false - иначе
* basePaymentMode (int) - режим базовой валюты(0 - валюту не меняли, subtotalByCard - переход в подытог по карте, paySum - оплата указанной суммой (остальные платежи базовой валютой))
* basePaymentSum (double) - сумма для оплаты базовой валютой
* payableSum (double) - сумма к оплате(сумма, которую необходимо доплатить)
* consultantName (string) - имя консультанта
* currentPosition (int) - индекс выделенной пользователем позиции
* frDocNum (string) - номер документа из ФР
* frDocCopy (string) - копия документа в ФР
* userName (string) - имя пользователя, создавшего документ
* customerAddress (string) - адрес(телефон или email) покупателя
* closeWithoutPrint (bool) - флаг закрытия документа без печати. true - закрывать документ без печати
* egaisCounter (int) - счетчик успешных регистраций в ЕГАИС
* backReasonCode (int) - код причины возврата
* actorCode (string) - код пользователя, подтверждающего некоторое действие
* sourceIdentifier (string) - идентификатор исходного документа при возврате
* id (int) - идентификатор записи документа из БД
* workShiftId (int) - идентификатор записи смены для текущего документа из БД
* num (int) - номер чека(тот самый, что участвует в документообороте и отображается в GUI)
* type (int) - тип документа (0 - не определен, 1 - чек продажи, 2 - чек возврата, 3 - акт внесения денег, 4 - акт выема денег, 5 - поступление товара, 6 изъятие товара из торговли, 7 - отмена чека продажи, 8 - отмена чека возврата,  
  9 - отмена внесения денег, 10 - отмена изъятия денег, 11 - отмена поступления товара, 12 - отмена изъятия товара, 13 - остаток денег на начало смены, 14 - инкассация денег, 15 - отмена инкассации денег,  
  16 - инвентаризация, 17 - возврат тары, 18 - возврат поставщику, 19 - заказ, 20 - режим обучения контрольным весам, 25 - чек возврата по чеку продажи, 26 - чек коррекции прихода, 27 - чек коррекции расхода)
* dept (int) - номер отдела
* cashCode (string) - код кассы
* shift (int) - номер смены документа
* cashierCode (string) - код кассира
* dateTimeBeg (datetime) - время создания(открытия) документа
* dateTimeEnd (datetime) - время закрытия документа
* sum1 (double) - сумма документа без скидок
* sum2 (double) - сумма документа без скидок
* sumb (double) - итоговая сумма документа
* discountBase (double) - ???
* sumcash (double) - сумма наличности в документе
* addingAllowed (bool) - флаг возможности добавления позиций в документ. true - можно добавлять позиции, false - нельзя
* status (int) - статус документа(0 - отсутствует, 1 - в режиме формирования, 2 - в режиме оплаты, 3 - в режиме сдачи, 4 - закрыт, 5 - отменен, 6 - отложен, 7 - в режиме заказа)
* databaseStatus (int) - статус документа в БД(1 - закрыт, 2 - отменён, 0 - остальное)
* linkedDocumentId (string) - идентификатор связанного документа
* rtext (string) - текст для печати на чековой ленте
* positionsCount - количество позиций в документе
* waresPositionsCount - количество позиций в документе, исключая тару
* waresQuantity - сумма количества штучных позиций, исключая тару и считая, что каждая весовая позиция считается как одна штука
* quantity (double) - суммарное количество товаров в документе
* netto (double) - сумма количества весового товара без тары
* brutto (double) - сумма количетсва весового товара
* departmentCode (int) - номер отдела
* departmentName (string) - имя отдела
* departmentClientItemId (int) - идентификатор записи клиента отдела из БД
* departmentDocumentPrefix (string) - префикс документа для отдела
* shopOptions (string) - получение extendedoptions из БД таблицы shop
* shopLabels (string) - получение меток магазина
* shopCode (string) - получение кода магазина
* changeSum (double) - сумма сдачи(сумма оплаты - сумма документа)
* paymentSum (double) - сумма оплат в документе
* discWithoutBonus (double) - сумма дисконта без учёта бонусов
* discBonus (double) - сумма бонусов как скидок
* maxSum (double) - максимальная сумма документа
* considerRemain (bool) - флаг учёта остатков. true - учитывать остатки при продаже
* unitePosition (string) - настройка объединения позиций("disable", "all", "piece")
* positionsLimit (int) - лимит количества позиций в документе
* uniqueIdPattern (string) - паттерн образования уникального идентификатора документа
* deptMode (string) - настройка разнемения сумм по отделам в возврате по чеку продажи("deny", "logicalDept", "frDept")
* percentDiscount (double) - ???
* absoluteDiscount (double) - ???
* newCheck (bool) - флаг того, что документ является новым. true - новый документ
* lastValutCode (long int) - ???
* backMode (bool) - флаг возврата в чеке продажи. true - возврат в чеке продажи
* wasDeferred (bool) - флаг того, что документ был отложен. true - документ отложен
* dateincrement (int) - смещение рабочей даты (производство)
* priceType (int) - ???
* valut (int) - валюта в документе(не используется)
* operationId (?) - ?
* deferredSent (bool) - флаг того, что отложенный чек был отправлен на сервер. true - отправлен на сервер
* currentPayableSum (double) - сумма наличной оплаты которая выполняется в данный момент
* currentPaymentSum (double) - сумма оплат в документе + сумма наличной оплаты которая выполняется в данный момент
* countersStatus (int) - статус счётчиков дисконта(0 - счетчики отключены, 1 - ошибка при получении счетчиков, 2 - счетчики получены)
* countersSourceCardNumber(string) - номер карты по которой сработали счётчики дисконта
* personalPricesCardNumber(string) - номер карты по которой сработали персональные скидки
* fiscalIdentifier (string) - фискальный идентификатор документа
* correctionReasonCode (int) - код основания документа коррекции
* correctionSourceDocumentNumber (string) - идентификатор исходного документа для коррекции
* correctionSourceDocumentDate (date) - не поддежано логикой форматирования параметров кассы
* correctionType (int) - тип коррекции(0 - самостоятельная операция, 1 - операция по предписанию)
* waybillPrintRequireStatus (int) - статус печати счёт-фактуры(0 - не определено, 1 - печать счета-фактуры не требуется, 2 - печать счета-фактуры требуется)
* version (string) - версия кассы
* waybillNumber (string) - номер счёт-фактуры

cardRecord:

* id (int) - идентификатор записи в БД
* inputSource (int) - источник ввода карты(0 - не определено, 1 - вручную, 2 - сканер штрих-кода, 4 - ридер магнитных карт, 8 - банковский терминал, 16 - получение данных с удалённого сервера)
* dateTime (datetime) - дата создания карты(добавления карты в чек)
* bonusBalance (double) - баланс бонусных баллов на карте
* position (int) - номер позиции карты в документе
* codeinternal (int) - внутренний номер карты в документе
* clientItemId (int) - идентификатор записи клиента из БД
* idCard (int) - идентификатор записи карты из справочника
* number (string) - номер карты
* cardSum (double) - сумма карты
* blocked (bool) - флаг того, что карта заблокирована. true - заблокирована
* idCardGroup (int) - идентификатор записи группы карт в БД
* cardGroupName (string) - имя группы карт
* cardGroupText (string) - описание группы карты
* cardMode (int) - вид лояльности (0 - дисконтная, 1 - Артикс Бонус, 2 - Артикс:Дисконт, 4 - Карта БМС, 5 - Карта Террасофт, 6 - Социальная карта, 7 - RSLoyalty, 8 - CardLab, 9 - Золотая корона, 10 - EPS,  
  11 - Loymax, 12 - ABMLoyalty)
* inputModeMask (int) - способ ввода карт из группы(0 - не определено, 1 - вручную, 2 - сканер штрих-кода, 4 - ридер магнитных карт, 8 - банковский терминал, 16 - получение данных с удалённого сервера)
* regexPattern (string) - маска номеров карт (регулярное выражение)
* regexRule (string) - правило преобразования номера карты (регулярное выражение)
* valutCode (int) - код валюты (устанавливается для бонусных карт, не используется)
* prefixBeg (string) - начало диапазона префикса номеров карт
* prefixEnd (string) - конец диапазона префикса номеров карт
* discountTypeInt (int) - тип скидки (0 - процентная, 1 - абсолютная, 2 - индексная)
* discountValue (double) - значение скидки
* lenBeg (int) - начало диапазона длин номеров карт
* lenEnd (int) - конец диапазона длин номеров карт
* multiplicator (double) - мультипликатор карты
* multiplicatorCurrent (double) - текущий мультипликатор (по умолчанию)
* multiplicatorBeg (datetime) - дата начала действия мультипликатора
* multiplicatorEnd (datetime) - дата окончания действия мультипликатора
* bonusBalanceInactive (double) - сумма неактивных средств на карте
* pointsForSpend (double) - сумма баллов для списания
* pointsForEarn (double) - сумма баллов для начисления
* rawData (string) - номер карты без преобразований
* purchases (int) - количество покупок
* terminalId (string) - id терминала, EPS Token, BMS
* operationId (string) - id операции, RRN, EPS CLID
* cardStatus (int) - статус карты (0 - анонимная, 1 - активная, 2 - заблокирована)
* verificationType (int) - идентификатор записи верификации из БД
* verificationStatus (int) - статус верификации(0 - не верифицирована, 1 - верифицирована)
* verificationAttribute (int) - признак верификации(0 - безусловная верификация, 1 - отложенная верификация, 2 - автоверификация)
* verificationMethodType (int) - тип верификации(0 - не выбрана, 1 - неизвестный тип верификации, 2 - верификация не требуется, 3 - автоверификация, 4 - СМС, 5 - Push уведомление, 6 - CardsMobile)
* verificationMethodName (string) - имя метода верификации
* verificationTypeName (stirng) - имя типа верификации
* hash (string) - хеш карты
* minPointsForSpend (double) - минимальная сумма для списания

client:

* clientitemid (int) - идентификатор записи клиента из БД
* idClient (string) - идентификационный номер клиента
* name (string) - имя клиента
* text (string) - текст(заметки)
* sex (int) - пол(0 - мужской, 1 - женский)
* specialDate1Name (string) - название специальной даты 1
* specialDate2Name (string) - название специальной даты 2
* specialDate3Name (string) - название специальной даты 3
* zipCode (string) - почтовый индекс
* address (string) - адрес
* email (string) - адрес электронной почты
* webPage (string) - веб страница
* phoneNumber (string) - номер телефона
* inn (string) - ИНН
* document (string) - документ
* okpo (string) - ОКПО
* okpd (string) - ОКРД
* occupation (string) - территория
* extendedOptions (string) - дополнительные опции
* childrenCount (int) - количество детей
* organizationCode (string) - уникальный код организации

position:

* posnum (int) - номер позиции в документе
* dept (int) - номер отдела
* item\_datetime (datetime) - дата создания позиции
* opcode (int) - код операции(0 - не определён, 50 - продажа, 51 - сторно продажи, 52 - продажа суммой в отдел, 53 - сторно продажи суммой в отдел, 54 - возврат в чеке продажи, 55 - сторно возврата в чеке продажи,  
  56 - возврат суммой в чеке продажи, 57 - сторно возврата суммой в чеке продажи, 58 - возврат, 59 - сторно возврата, 60 - возврат суммой в отдел, 61 - сторно возврата суммой в отдел,  
  62 - возврат тары, 63 - продажа сертификата, 64 - возврат сертификата, 70 - аннулирование продажи, 71 - аннулирование возврата, 200 - инвентаризация, 201 - инвентаризация суммой,  
  250 - возврат поставщику, 260 - заказ, 1000 - "внешний" платеж в чеке продажи, 1002 - "внешний" платеж в чеке возврата, 1004 - комиссия к "внешнему" платежу в чеке продажи,  
  1006 - комиссия к "внешнему" платежу в чеке возврата, 1010 - режим обучения контрольным весам, 1011 - превышение номинальной стоимости подарочной карты)
* bcode (string) - штрих-код
* name (string) - наименование позиции
* articul (string) - артикул
* scode (string) - код кассира (Какой? Который добавил, изменил или что-то другое?)
* additionaldata (string) - дополнительные данные
* extendetoptions (string) - дополнительные опции
* measure (int) - код единицы измерения
* bcode\_mode (int) - способ добавления позиции(0 - Количество по умолчанию, 1 - Ручной ввод кода, 2 - Код товара введен сканером, 3 - Код введен с помощью горячей клавиши, 4 - Код товара выбран из справочника,   
  5 - Товар введен с помощью "быстрого" кода, 6 - Товар введен с как составная часть комплекта, 7 - Товар введен внешней процедурой, 8 - Товар введен с ридера магнитных карт, 9 - Загружен из софт-чека)
* bquant (double) - количество
* bquant\_mode (int) - способ ввода количества(1 - Принято количество товара по умолчанию, 2 - Количество товара введено вручную, 3 - Количество выделено из отсканированного штрих-кода, 4 - Количество получено с весов,  
  5 - Количество задано внешней процедурой, 6 - Количество взято из софт-чека, 7 - Количество рассчитано по вместимости тары)
* ostat\_modif (int) - ???
* ostat\_mode (int) - не используется
* code (string) - код товара
* cquant (double) - количество в учетных единицах
* pricetype (int) - номер ценовой схемы
* pricevcode (int) - код типа оплаты
* price (double) - цена
* min\_price (double) - минимальная цена
* enteredPrice (double) - цена, введённая вручную
* hasEnteredPrice (bool) - флаг того, что цена была введена вручную
* hasDiscountPrice (bool) - флаг того, что цена получена из дисконта
* packingMode (bool) - флаг применения модификатора расфасовка. true - товар был расфасован
* price\_mode (int) - способ ввода цены(1 - цена получена из товара, 2 - цена введена вручную, 4 - цена выбрана из массива цен, 8 - использована отложенная цена, 16 - цена получена из акцизной марки)
* index (int) - ???
* index\_price (double) - индексная цена
* disc\_perc (double) - ???
* disc\_abs (double) - сумма скидки
* sumi (double) - начальная сумма до применения скидок
* sumb (double) - итоговая сумма после применения всех скидок
* sumn (double) - начальная сумма до применения скидок
* sume (double) - начальная сумма до применения скидок
* tara\_default (string) - ???
* groupId (string) - идентификатор группы(заполняется после применения модификатора группы)
* recalcable (bool) - флаг возможности перерасчёта позиции. true - позиция может быть перерасчитана
* itemId - идентификатор записи позиции в БД
* opid (int) - ???
* impactText (string) - "" - если нет скидок, "СО СКИДКОЙ" - если есть скидка, "С НАДБАВКОЙ" - если есть надбавка
* impactPercent (double) - процент скидки на позицию((цена\_без\_скидки - итоговая\_цена) / цена\_без\_скидки \* 100)
* impactSum (double) - сумма скидки позиции(цена\_без\_скидки - итоговая\_цена)
* bonusImpactSum (double) - сумма бонусного воздействия в виде списания баллов
* departmentCode (int) - код отдела
* departmentName (string) - наименование отдела
* departmentClientItemId (int) - идентификатор записи клиента отдела из БД
* departmentDocumentPrefix (string) - префикс документа для отдела
* tmcPrice (double) - цена из товара
* tmcTaraMode(int) - режим тары(0 - тара не используется, 1 - тара, 2 - отпускается только в тару, 3 - товар - карта, 4 - товар был/будет проверен в ЕГАИС, 5 - товар без акцизной марки был/будет проверен в ЕГАИС, 6 - услуга)
* tmcTaraCapacity (double) - вместимость тары
* vatCode1 (int) - код налога 1
* vatRate1 (double) - процент налога 1
* vatSum1 (double) - сумма налога 1
* vatCode2 (int) - код налога 2
* vatRate2 (double) - процент налога 2
* vatSum2 (double) - сумма налога 2
* vatCode3 (int) - код налога 3
* vatRate3 (double) - процент налога 3
* vatSum3 (double) - сумма налога 3
* vatCode4 (int) - код налога 4
* vatRate4 (double) - процент налога 4
* vatSum4 (double) - сумма налога 4
* vatCode5 (int) - код налога 5
* vatRate5 (double) - процент налога 5
* vatSum5 (double) - сумма налога 5
* unitCode (int) - код единицы измерения
* unitName (string) - наименование единицы измерения
* unitFlag (bool) - флаг весовой единицы измерения. true - весовая единица измерения, false - штучная
* softCheckId (string) - идентификатор софт-чека
* hasDiscount (bool) - флаг того, что позиция содержит дисконт
* bonusImpactEarn (double) - сумма бонусов для начисления
* aspectSchemeName (string) - наименование схемы разрезов
* aspectValueSetName (string) - наименование полного значения разреза
* allowDiscount (bool) - флаг, разрешающий применение дисконта. true - разрешено
* allowOnlyExternalDiscount (bool) - флаг, разрешающий применение только внешнего дисконта(внешние системы лояльности). true - разрешено
* allowBonusEarn (bool) - флаг, разрешающий начисление бонусов на позицию. true - разрешено
* allowBonusSpend (bool) - флаг, разрешающий списание бонусов. true - разрешено
* exciseMark (string) - акцизная марка
* additionalExciseMark (string) - серия и номер акцизной марки (дополнительный штрих-код DataMatrix акцизной марки)
* discName (string) - название дисконтной акции
* tags (string) - список тегов позиции
* inn (string) - ИНН
* kpp (string) - КПП
* alcoholPercent (double) - процент алкоголя
* alcTypeCode (int) - код вида алкогольной продукции
* paymentItemId (int) - идентификатор записи оплаты в БД
* consultantName (string) - имя консультанта
* additionalBarcode (string) - дополнительный штрих-код акцизной марки
* alcoCode (string) - алкокод
* bquantFromBcode (double) - количество из позиции
* packingPrice (double) - цена для расфасовки
* visualVerified (bool) - флаг того, что товар требует визуального подтверждения
* paymentObject (int) - признак предмета расчета
* paymentMethod (int) - признак способа расчета(определяется на основе признака предмета расчета. Если признак предмета расчета = 0, то признак способа расчета также 0.  
  Если признак предмета расчета = 10(аванс, задаток, предоплата, бонус и т.п), то признак способа расчета = 3(аванс). Для любого иного признака предмета расчета признак способа расчета = 4(полная оплата))
* tmcGroup (string) - название группы, к которой относится товар
* priceSource (int) - источник цены позиции(0 - из товара, 1 - введена вручную, 2 - выбрана из списка, 3 - цена из акцизной марки)