

BONEX · LOYALTY PLATFORM

Bonex — Integrations API

Документация по API для интеграции

Версия от 2026-06-18 · api.bonex.one

Bonex — API для интеграций (1С, Frontol)

Спецификация Cashier API для разработчиков кассового ПО и внедренцев 1С.

Базовый URL: `https://api.bonex.one/api/v1` (или ваш `BONEX_API_URL`).

1С — пошаговая инструкция для новичка

Полный текст с фрагментами кода форм — в скачанном BSL-файле (**Раздел А**). Ниже — краткий чеклист.

Шаг 0. Подготовка в Bonex (panel.bonex.one)

Действие	Где
Создать филиал (магазин)	Филиалы
Создать отдельную кассу на каждое РМК	Филиалы → Кассы
Скопировать Tenant Token	Интеграции → Учетные данные API
Для каждой кассы: Cashier Token + ID кассы (число)	Интеграции → выберите кассу в списке
Скачать модуль для вашей конфигурации	Интеграции → 1С → УНФ / Розница / УТ

ID кассы — это числовой идентификатор записи «Касса» в Bonex (не код в 1С). Виден в panel при выборе кассы на странице Интеграции и в разделе **Кассы**.

Шаг 1. Расширение в 1С

- Конфигуратор → режим с **разрешёнными расширениями**.
- Расширения → **BonexLoyalty** → создать.
- Константы (только общие на всю базу):
 - `BonexAPIURL` — URL API (`https://api.bonex.one/api/v1`)
 - `BonexTenantToken` — из panel
 - `BonexРежимСписанияБонусов` — `payment_type`
 - `BonexВидОплатыБонекс` — `бонусы`
 - `BonexАвтоРегистрация` — `Истина`

Не создавайте константы `BonexCashierToken` и `BonexCashierNodeID` — токен кассы хранится **отдельно на каждом ПК**.

- Заимствовать документы → добавить реквизиты `BonexТелефонКлиента` ,
`BonexСуммаСписанияБонусов` :
 - УНФ: `ЧекККМ` , `РасходнаяНакладная`
 - Розница: `ЧекККМ`
 - УТ: `ЧекККМ` , `РеализацияТоваровУслуг`

- Общий модуль `ВопехИнтеграция` — код из **Раздела В** BSL-файла.
- Подписки на **ОбработкаПроведения** для тех же документов.

Шаг 2. Регистрация каждой РМК (обязательно)

На **каждом** компьютере с кассой, один раз:

```
ВопехИнтеграция.УстановитьТокенКассыВопех("<Cashier Token из panel>", <ID кассы>);
```

- Cashier Token** — уникальный для этой кассы (panel → Интеграции → выбрать кассу).
- ID кассы** — число из panel (та же касса).

Проверка на этом ПК:

```
ВопехИнтеграция.ПроверитьНастройкуКассыВопех()
```

Для администратора без консоли: создайте в расширении обработку **ВопехНастройкаКассы** (форма с полями токен + ID, кнопка «Сохранить» вызывает `УстановитьТокенКассыВопех`). Подробно — **A7** в BSL.

Токен сохраняется в `ХранилищеСистемныхНастроек` по имени компьютера (общий для всех пользователей 1С на этом ПК). **Не копируйте** один токен на все РМК.

Старые версии модуля писали в `ХранилищеОбщихНастроек` (привязка к пользователю) — кассир не видел токен администратора. Обновите модуль `ВопехИнтеграция` и повторите `УстановитьТокенКассыВопех`.

Шаг 3. РМК — удобный сценарий для кассира

На форме **РабочееМестоКассира** → **ФормаРМК** (см. **A8** в BSL):

Элемент	Назначение
Вопех — клиент	QR / телефон → баланс → «сколько списать?» (Enter = максимум)
Вопех — списать макс.	Без второго вопроса — сразу максимум бонусов
Вопех — сброс	Новый чек, убрать клиента и бонусы
ВопехСтрокаСтатус	Надпись на форме: «к оплате XXX сом»

Алгоритм для кассира (3 шага):

- Набрать товары.
- Вопех — клиент** → отсканировать или ввести телефон → Enter (или указать сумму бонусов).
- Оплата** → внести только **остаток** наличными и/или картой → пробить чек.

Модуль сам добавляет строку «Бонусы (Вопех)» в оплату. Кассиру **не нужно** вручную выбирать вид оплаты `Вопех`.

РМК: только `ПоказатьВводСтроки`, не `ВвестиСтроку`. Готовый код формы — **A8** в BSL.

Автоскан: в обработчике сканера:

```
Если WopecИнтеграция.ЭтоИдентификаторКлиентаWopec(Штрихкод) Тогда
    WopecОбработатьСканНаСервере(Штрихкод); // см. А8
    Возврат;
КонецЕсли;
```

Перед пробитием: перехват `ЗаписатьЧекККМПередПробитием` →
`WopecСкопироватьРеквизитыРМКВЧек(Форма)` (шаг 6 в А8).

Крупная надпись «К ОПЛАТЕ»: реквизит `WopecСтрокаСтатус` + панель на форме — пошагово **А8а** в BSL (автоматически через `WopecРазместитьПанельСтатуса` или вручную в конфигураторе).

Шаг 4. Расходная накладная / Реализация (опт)

На **форме документа** (заимствовать `ФормаДокумента`):

- Кнопки **Wopec: телефон** и **Wopec: карта / QR** (как в РМК).
- Вызов: `WopecИнтеграция.СканироватьКлиентаWopec(Идентификатор, Объект, Объект.СуммаДокумента)`.
- Выведите реквизит `WopecТелефонКлиента` на форму (просмотр после скана).

Фрагменты кода — **А9** в BSL (УНФ; для УТ — `РеализацияТоваровУслуг` вместо `РасходнаяНакладная`).

Шаг 5. Включить и проверить

1. Обновить конфигурацию БД → включить расширение в режиме Предприятия.
2. На каждом ПК РМК — шаг 2.
3. Тест: кнопка Wopec → телефон клиента из panel → **провести** чек → баланс изменился в panel.

Типичные ошибки

Сообщение	Решение
«Токен кассы не задан»	Выполнить шаг 2 на этом ПК
HTTP 401/403	Неверный Tenant Token или Cashier Token; повторить шаг 2
«Пропуск: не указан телефон»	Не нажали Wopec до проведения; скан/ввод телефона обязателен
HTTP 404	Клиента нет в panel (или включить <code>WopecАвтоРегистрация</code>)

Аутентификация

Все запросы кассы:

```
X-Tenant-Token: <64-символьный токен магазина>
X-Cashier-Token: <токен cashier node – одна касса / точка>
X-Device-Id: <идентификатор РМК / терминала – обязателен>
Content-Type: application/json
Accept: application/json
```

Привязка к устройству (защита лицензии)

При **первом** запросе с валидной парой токенов Bonex запоминает `X-Device-Id` за этой кассой. Все следующие запросы с тем же `X-Cashier-Token` должны приходить с **тем же** `X-Device-Id`, иначе **403**.

Система	Значение X-Device-Id
1С УНФ / Розница / УТ	<code>КлючРабочегоМестаBonex()</code> — <code>ИмяКомпьютера()</code> (автоматически в модуле)
Frontol	<code>GetTerminalId()</code> — serial кассы или имя ПК (в скрипте)
cash.bonex.one	<code>web-<uuid></code> — генерируется при первом открытии, хранится в <code>localStorage</code> браузера

Перенос кассы на новый ПК: panel → **Кассы** → «Сбросить привязку» (или ротация токена).

Суперадмин: **admin** → **Магазин** → **Активность касс** → **Сбросить привязку**.

Попытки с чужого устройства увеличивают `device_mismatch_count` и видны в админке.

Токены: **panel** → **Интеграции** → **Учетные данные API**.

Лимит: **300 запросов/мин** на кассу (`throttle:cashier`).

Два токена — зачем

Заголовок	Уровень	Сколько
<code>X-Tenant-Token</code>	Весь магазин в Bonex (договор, клиентская база)	1 на магазин
<code>X-Cashier-Token</code>	Одна оплаченная касса / РМК	1 токен = 1 касса

Токены задаются в **panel.bonex.one** → **Интеграции** (после создания касс в **Филиалы** → **Кассы**).

1С УНФ на сервере + несколько РМК

Типичная схема: одна информационная база на сервере, несколько рабочих мест РМК в разных магазинах.

Параметр	Где хранится
<code>BonexTenantToken</code> , <code>BonexAPIURL</code>	Константы 1С — один раз на всю базу
Cashier Token + ID кассы	ХранилищеСистемныхНастроек — на каждом ПК через <code>УстановитьТокенКассыBonex</code>

Константы `BonexCashierToken` и `BonexCashierNodeID` **не используются** — не создавайте их в расширении.

На **каждом** рабочем месте РМК один раз:

```
BonexИнтеграция.УстановитьТокенКассыBonex("<Cashier Token из panel>", <ID кассы>);
```

- **Cashier Token** — panel → Интеграции → выберите кассу этого ПК.

- **ID кассы** — число той же записи в panel (не путать с номером кассы ККМ в 1С).

Проверка: `BonexИнтеграция.ПроверитьНастройкуКассыBonex()` → Настроено = Истина .

Без шага регистрации API вернёт ошибку «Токен кассы не задан» (не HTTP-запрос с пустым токеном).

Суперадмин: **admin.bonex.one** → **Магазин** → «**Активность касс**» (ring, продажи, признак «шаринга» токена).

Формат QR клиента

QR в Mini App и Flutter кодирует **телефон в международном формате**:

```
+992901234567
```

2D-сканер на кассе передаёт эту строку как штрихкод. Отдельный «зашифрованный токен» не используется — идентификатор клиента = нормализованный телефон в Bonex.

Нормализация на стороне кассы (рекомендуется):

- убрать пробелы, скобки, дефисы;
- если нет `+`, добавить `+992` для 9-значных номеров Таджикистана.

Простой сценарий для кассира и менеджера

Один шаг: отсканировать QR / штрихкод карты **или** ввести телефон — клиент привязан к чеку/ документу.

Система	Что вызвать	Когда
1С РМК	<code>СканироватьКлиентаBonex(Штрихкод, ДокументЧека, Сумма)</code>	При скане на кассе
1С опт (УНФ / УТ)	<code>СканироватьКлиентаBonex(Штрихкод, Объект, Сумма)</code>	Кнопка на форме накладной
Frontol	<code>OnBonexBarcodeScanned(barcode)</code> или <code>OnBonexPhoneEntered(phone)</code>	Сканер / кнопка телефона

Принимается один и тот же идентификатор: QR (телефон), EAN-13 (`200 ...`), клубная карта (`001-0000123`), телефон.

Авторегистрация: если клиента нет в Bonex, но введён телефон — создаётся с кассы (`POST /cashier/client/register`). В 1С: константа `BonexАвтоРегистрация = Истина` . В Frontol: `BONEX_AUTO_REGISTER = true` .

После скана кассир видит: имя, баланс, лимит списания. При проведении документа бонусы начисляются автоматически.

Сценарий интеграции (API)

Шаг 1 — Скан QR / штрихкод / ввод телефона

```
GET /cashier/client/{identifier}?check_amount={сумма_чека}
```

Параметр	Описание
identifier	URL-encoded телефон (%2B992 ...), EAN-13, клубная карта
check_amount	Опционально. Текущая сумма открытого чека — для расчёта max_redeem_amount

Ответ 200:

```
{
  "success": true,
  "data": {
    "id": 42,
    "phone": "+992901234567",
    "full_name": "Парвиз Солиев",
    "bonus_balance": 150,
    "redeemable_balance": 120,
    "held_balance": 30,
    "welcome_bonus_balance": 0,
    "total_spent": 1200,
    "tier": { "id": 2, "name": "Gold", "color": "#C9A227" },
    "is_blocked": false,
    "loyalty": {
      "bonus_percent": 5,
      "max_redeem_percent": 30,
      "redeem_hold_seconds": 86400,
      "redeem_hold_human": "24 ч",
      "redeemable_balance": 120,
      "held_balance": 30,
      "min_purchase_amount": 10,
      "currency": "TJS",
      "max_redeem_amount": 60
    }
  }
}
```

Правила списания:

- `bonus_balance` — весь баланс клиента.
- `redeemable_balance` — **доступно к списанию сейчас**: бонусы, прошедшие выдержку `redeem_hold_seconds` (по умолчанию **86400 сек = 24 часа** после начисления; на демо-базе — **1 секунда**).
- `redeem_hold_human` — та же выдержка человекочитаемо («24 ч», «1 сек», «180 дн»).
- `held_balance` — бонусы «на выдержке» (начислены недавно), списать нельзя.
- `max_redeem_amount` присутствует только если передан `check_amount > 0`.

Формула: `max_redeem_amount = min(check_amount × max_redeem_percent / 100, redeemable_balance)`.

Процент (например 30%) применяется **только к сумме чека**, а не к балансу.

Пример: чек 500, лимит 30% = 150; баланс 40 (доступен) → списать можно все **40**.

Пример: чек 500, лимит 30% = 150; доступно 200 → списать можно максимум **150**.

Ошибки: `404` клиент не найден, `403` заблокирован.

UI кассы: показать имя, баланс, **доступно к списанию** и сколько на выдержке, предложить списание до `max_redeem_amount`.

Шаг 2 — Закрытие чека (начисление + списание)

```
POST /cashier/sale
```

Режим «Вид оплаты» (рекомендуется, `redeem_mode: payment_type`)

Бонусы — отдельная строка в окне оплаты РМК (как подарочный сертификат). Сумма чека в 1С **не уменьшается**.

Пример: чек 1000 сомони, списано 45 бонусов, к оплате деньгами 955.

```
{
  "client_phone": "+992901234567",
  "total_amount": 1000.00,
  "bonus_redeem_amount": 45.00,
  "redeem_mode": "payment_type",
  "external_sale_id": "a1b2c3d4-e5f6- ... ",
  "cashier_node_id": 1,
  "sale_date": "2026-06-05T14:30:00"
}
```

Режим «Скидка» (`redeem_mode: discount`)

1С пропорционально уменьшает цены в строках чека. В API передаётся итог **после скидки** и полная сумма:

```
{
  "client_phone": "+992901234567",
  "total_amount": 955.00,
  "gross_amount": 1000.00,
  "bonus_redeem_amount": 45.00,
  "redeem_mode": "discount",
  "external_sale_id": "a1b2c3d4-e5f6- ... "
}
```

Если `gross_amount` не передан в режиме скидки, бэкенд вычисляет его как `total_amount + bonus_redeem_amount`.

База начисления кэшбэка (panel → Программа лояльности)

Настройка <code>accrual_base</code>	<code>payment_type</code> (чек 1000, бонусы 45)	<code>discount</code> (чек 955, бонусы 45)
<code>net_cash</code> (по умолчанию)	5% от 955	5% от 955
<code>gross</code>	5% от 1000	5% от 1000

Полный пример с строками чека

```
{
  "client_phone": "+992901234567",
  "total_amount": 200.00,
  "bonus_redeem_amount": 50.00,
  "redeem_mode": "payment_type",
  "external_sale_id": "a1b2c3d4-e5f6- ... ",
  "cashier_node_id": 1,
  "cashier_id": null,
  "payment_method": "card",
  "sale_date": "2026-06-05T14:30:00",
  "lines": [
    {
      "sku": "SKU-001",
      "category_code": "DAIRY",
      "name": "Молоко 1л",
      "quantity": 2,
      "unit_price": 20.00,
      "line_amount": 40.00
    }
  ]
}
```

Поле	Обязательно	Описание
<code>client_phone</code>	да	Телефон клиента
<code>total_amount</code>	да	Итог чека в 1С: полная сумма (<code>payment_type</code>) или после скидки (<code>discount</code>)
<code>gross_amount</code>	нет	Полная сумма до скидки бонусами (для <code>discount</code> ; опционально)
<code>bonus_redeem_amount</code>	нет	Сумма бонусов к списанию до начисления
<code>redeem_mode</code>	нет	<code>payment_type</code> (по умолчанию) или <code>discount</code>
<code>external_sale_id</code>	нет	UUID/номер чека в 1С — идемпотентность
<code>receipt_number</code>	нет	Номер фискального чека для истории покупок клиента
<code>cashier_node_id</code>	нет	По умолчанию — <code>node</code> из <code>X-Cashier-Token</code>
<code>lines</code>	нет	Строки чека: <code>name</code> , <code>quantity</code> , <code>unit_price</code> , <code>line_amount</code>

Порядок на бэкенде: создать `Sale` → `redeem` (если `bonus_redeem_amount` > 0) → `accrueFromSale` → рефералы, реактивация, `second-purchase`.

Ответ 201:

```
{
  "success": true,
  "data": {
    "id": 1001,
    "total_amount": 200,
    "bonus_accrued": 7.5,
    "bonus_redeemed": 50,
    "new_balance": 107.5,
    "sale_date": "2026-06-05T14:30:00+00:00"
  }
}
```

Ошибки 422 (примеры):

- сумма покупки ниже `min_purchase_amount` ;
- `bonus_redeem_amount` > баланса или > лимита % чека;
- welcome-бонусы при чеке ниже `welcome_min_redeem_amount` .

Альтернатива — только списание

```
POST /cashier/redeem
```

```
{
  "client_phone": "+992901234567",
  "amount": 50,
  "total_amount": 200,
  "external_sale_id": " ... "
}
```

Обычно достаточно `POST /cashier/sale` с `bonus_redeem_amount` .

Возврат товара (1С → Вонех)

Двухшаговый сценарий: предрасчёт → проведение документа возврата в 1С → фиксация в Вонех.

Шаг 1 — Предрасчёт (хватает ли бонусов)

```
POST /cashier/return/calculate
```

```
{
  "client_phone": "+992901234567",
  "bonus_to_revoke": 50,
  "original_external_sale_id": "uuid-исходного-чека"
}
```

Если `bonus_to_revoke` не передан, но указан `original_external_sale_id` , Вонех возьмёт `bonus_accrued` из исходной продажи.

Ответ — бонусов хватает:

```
{
  "success": true,
  "data": {
    "status": "ok",
    "available_bonuses": 150,
    "bonus_to_revoke": 50
  }
}
```

Ответ — бонусов не хватает (не ошибка, а инструкция кассиру):

```
{
  "success": true,
  "data": {
    "status": "insufficient_bonuses",
    "available_bonuses": 0,
    "bonus_to_revoke": 50,
    "shortage_bonuses": 50,
    "shortage_cash_equivalent": 50
  }
}
```

1С показывает кассиру: уменьшить выдачу наличных на `shortage_cash_equivalent` (950 вместо 1000). Документ возврата в 1С остаётся на полную сумму товара.

Шаг 2 — Проведение возврата

```
POST /cashier/return
```

Обычный возврат (баланс \geq бонусов к аннулированию):

```
{
  "client_phone": "+992901234567",
  "return_amount": 1000,
  "cash_refund_amount": 1000,
  "original_external_sale_id": "uuid-исходного-чека",
  "external_return_id": "uuid-документа-возврата"
}
```

Возврат с нехваткой бонусов (клиент уже потратил начисленные бонусы):

```
{
  "client_phone": "+992901234567",
  "return_amount": 1000,
  "cash_refund_amount": 950,
  "shortage_settled": true,
  "original_external_sale_id": "uuid-исходного-чека",
  "external_return_id": "uuid-документа-возврата"
}
```

Логика бэкенда при `shortage_settled: true` :

1. Аннулирует все `bonus_to_revoke` (баланс может уйти в минус: $0 - 50 = -50$).

2. Начисляет `shortage_cash_equivalent` как урегулирование ($-50 + 50 = 0$).
3. Восстанавливает бонусы, списанные при исходной покупке (`bonus_redeemed`), если не указано иное.

Если нехватка есть, а `shortage_settled` не передан — **422** с телом:

```
{
  "success": false,
  "error": "Недостаточно бонусов для аннулирования при возврате.",
  "data": {
    "status": "insufficient_bonuses",
    "available_bonuses": 0,
    "bonus_to_revoke": 50,
    "shortage_bonuses": 50,
    "shortage_cash_equivalent": 50
  }
}
```

Ответ 201:

```
{
  "success": true,
  "data": {
    "id": 12,
    "return_amount": 1000,
    "cash_refund_amount": 950,
    "bonus_revoked": 50,
    "bonus_shortage": 50,
    "bonus_shortage_settled": 50,
    "new_balance": 0
  }
}
```

В оффере программы лояльности (Mini App при регистрации) укажите право удерживать сумму, эквивалентную уже использованным бонусам, из денежного возврата за товар.

Чек без клиента (KPI)

POST `/cashier/receipt`

```
{
  "total_amount": 150,
  "external_receipt_id": "uuid-чека",
  "cashier_node_id": 1,
  "receipt_date": "2026-06-05T14:30:00"
}
```

Регистрация на кассе (редко)

```
POST /cashier/client/register
{ "phone": "+992 ... ", "full_name": "Имя" }
```

Без реферального кода (в отличие от OTP-регистрации клиента).

Остатки (1С → Bonex)

Только X-Tenant-Token (без кассового токена):

```
POST /integration/inventory/sync
```

```
{
  "items": [
    { "sku": "SKU-001", "name": "Товар", "category_code": "DAIRY", "quantity": 12 }
  ],
  "deactivate_missing": true
}
```

Используется для персональных push по неликvidу (panel → Таргетинг).

Веб-касса (cash.bonex.one)

Панель кассира без 1С/Frontol — для малого бизнеса (телефон/планшет кассира).

Подключение

1. **panel** → **Филиалы** → **Кассы** — создайте кассу на **каждое** устройство (не копируйте один Cashier Token на два телефона).
2. **panel** → **Интеграции** → **Касса** — скопируйте Tenant Token, Cashier Token и откройте **Cashier App URL**.
3. Введите оба токена → «Сохранить и войти».
4. При первом запросе устройство **привязывается** к кассе (X-Device-Id отправляется автоматически).
5. В шапке отображаются **магазин, филиал и название кассы** (GET /cashier/session).

Один токен — одно устройство

Если кассир откроет тот же Cashier Token на другом телефоне → **403** «привязан к другому устройству».

Ситуация	Действие
Новый телефон вместо старого	panel → Кассы → Сброс привязки → войти снова на cash.bonex.one
Вторая касса в том же магазине	Создать новую кассу в panel → свой Cashier Token

API сессии

```
GET /cashier/session
X-Tenant-Token: ...
X-Cashier-Token: ...
X-Device-Id: web- ...
```

Ответ: `store_name`, `cashier_name`, `branch`, `cashier_node_id`, статус привязки.

Frontol ScriptEngine

Файл `frontol_bonex.js` (скачивается из **panel** → **Интеграции** → **Frontol**, с выбором кассы) уже содержит:

```
var BONEX_TENANT_TOKEN = "..."; // один на магазин
var BONEX_CASHIER_TOKEN = "..."; // уникальный для выбранной кассы
var BONEX_CASHIER_NODE_ID = 2;
```

Frontol 6.28.x (в т.ч. Demo): пошаговая установка —

`backend/resources/integrations/frontol/README.ru.md`.

В скачанном файле уже реализованы `HttpGet / HttpPost` (WinHttp), диалоги (VBScript), сумма чека (`frontol.currentDocument.sum`) и точки входа сценария.

Несколько терминалов Frontol

В отличие от 1С на сервере, отдельная процедура настройки токена **не нужна** — токен кассы зашит в скачанный файл.

Терминал	Действие
Касса 1	panel → выбрать «PMK-1» → скачать <code>frontol_bonex.js</code> → установить только на терминал 1
Касса 2	выбрать «PMK-2» → скачать другой файл → терминал 2

Не копируйте один и тот же `frontol_bonex.js` на все кассы — иначе все терминалы будут с одним `BONEX_CASHIER_TOKEN` (как оплата за 1 кассу при работе на нескольких). Контроль: **admin.bonex.one** → **Магазин** → **Активность касс**. При утечке — **Ротация** токена в panel → Кассы и новый скрипт на терминал.

Функция	Когда вызывать
<code>SearchCardByPhone()</code>	Кнопка / горячая клавиша — ввод телефона
<code>AttachCard()</code>	После скана карты Frontol
<code>EnterBonusPayment()</code>	Повторный ввод суммы списания
<code>BonexBeforeFiscalize()</code>	Перед пробитием — вызывает <code>BeforeCheckClose</code> → API sale
<code>ApplySumDiscountToPosition('bonexBonus')</code>	Маркетинговая акция «скидка из сценария»
<code>OnBonexBarcodeScanned(barcode)</code>	Программно, если раскладка передаёт штрихкод
<code>OnCheckOpened()</code> / <code>OnNewDocument()</code>	Новый чек (сброс клиента)

Заглушки HTTP/диалогов **не нужны** — они уже в скачанном скрипте для Frontol 6.

Псевдокод:

```
// При скане QR
OnBonexBarcodeScanned(scannedText); // → GET /cashier/client/{phone}

// При закрытии чека
BeforeCheckClose(total); // → POST /cashier/sale
```

1С — ключевые процедуры модуля

Скачайте BSL для вашей конфигурации (УНФ / Розница / УТ). **Раздел А** в файле — полная установка расширения.

Процедура	Назначение
УстановитьТокенКассыBonex(Токен, ID)	Регистрация кассы на текущем ПК (шаг А7)
ПроверитьНастройкуКассыBonex()	Проверка, задан ли токен на этом рабочем месте
ЭтоИдентификаторКлиентаBonex(Штрихкод)	Отличить QR/телефон/карту от штрихкода товара в РМК
ОбработатьКлиентаРМКBonex(...)	РМК: поиск + списание + строка оплаты + текст «к оплате XXX»
BonexСброситьКлиентаРМК(Док)	Сброс клиента и бонусов в текущем чеке
BonexСкопироватьРеквизитыРМКВЧек(Форма)	Перед пробитием: телефон и сумма бонусов в ЧекКМ
BonexКраткийСтатусРМК(Результат)	Короткая строка для панели на форме
BonexТекстПодсказкиРМК()	Текст панели до выбора клиента
СканироватьКлиентаBonex(Идентификатор, Док, Сумма)	Опт / накладная: поиск + привязка к документу
НайтиКлиентаВBonex(Идентификатор, СуммаЧека)	Только поиск (QR / EAN / карта / телефон)
ЗарегистрироватьКлиентаВBonex(Телефон, Имя)	Быстрая регистрация с кассы
ПривязатьКлиентаКДокументу(Док, Телефон)	Записать BonexТелефонКлиента на документ
ОтправитьПродажуВBonex(Док)	POST sale при проведении документа
ПолучитьСтрокиПродажиДляBonex(Док)	Автоматически: ТЧ → lines (товары в истории клиента)
ПолучитьНомерЧекаИзДокумента(Док)	Номер чека → receipt_number
ПолучитьСуммуОплатыBonexИзДокумента(Док)	Сумма оплаты видом «Бонусы (Bonex)»
РассчитатьВозвратВBonex(Телефон, Бонусов, ИдЧека)	POST /cashier/return/calculate
ОтправитьВозвратВBonex(Док, ...)	POST /cashier/return при проведении возврата
ОтправитьОстаткиВBonex()	Регламент inventory sync

Списание бонусов в 1С

Рекомендуемый режим — «Вид оплаты» (BonexРежимСписанияБонусов = payment_type):

1. Справочник «Виды оплат» → элемент «Бонусы (Bonex)» + включить в настройках РМК.
2. Кассир: **Bonex — клиент** (А8) → модуль сам ставит строку бонусов.
3. Кассир: **Оплата** → только остаток (нал + карта).

Ручное разбиение оплаты нужно только если не используете сценарий А8.

Альтернатива — «Скидка» (`discount`): 1С уменьшает цены в строках; в API уходит `total_amount` после скидки и `gross_amount` .

Реквизиты расширения на документе чека (опционально):

- `ВонехТелефонКлиента` — после скана QR;
- `ВонехСуммаСписанияБонусов` — запасной путь списания;
- `ВонехНомерЧека` — если номер не в стандартном поле `Номер` .

Строки чека выгружаются автоматически из ТЧ `Запасы` / `Товары` / `ТоварыИУслуги` : наименование, количество, цена, сумма, артикул, категория.

Подписки на события:

Конфигурация	Документы
УНФ 3.0	ЧекККМ, РасходнаяНакладная
Розница 3.0	ЧекККМ
УТ 11	ЧекККМ, РеализацияТоваровУслуг

Идемпотентность

Повторный `POST /cashier/sale` с тем же `external_sale_id` → **422** «Продажа с таким `external_sale_id` уже зарегистрирована».

Рекомендация: использовать `УникальныйИдентификатор()` ссылки документа 1С или номер фискального чека.

Уведомления после sale

Асинхронно (очередь Redis):

- Telegram `sendMessage` — если есть `telegram_user_id` ;
- FCM push — если клиент регистрировал `device token` в приложении.

Кассовое ПО уведомления не отправляет — только `Вонех API`.

Сгорание бонусов и напоминания

- Бонусы сгорают через `bonus_expiry_days` после начисления (демо-база: **180 дней**).
- За `bonus_expiry_reminder_days` дней до сгорания клиент получает напоминание: **Push** (если есть приложение) → иначе **Telegram** → иначе **SMS** (если у магазина настроен `OsonSMS`).
- Команды (по расписанию): `bonex:expiring-bonus-reminders` (09:30) и `bonex:expire-bonuses` (03:00).
- Выдержка перед списанием (`redeem_hold_seconds`) и срок сгорания (`bonus_expiry_days`) настраиваются в `panel` → Лояльность. Интеграция 1С/Frontol ничего дополнительно не делает — лимиты приходят в ответе `lookup`.

Тестирование

```
# Lookup
curl -s -H "X-Tenant-Token: ..." -H "X-Cashier-Token: ..." \
  "https://api.bonex.one/api/v1/cashier/client/%2B992900000500?check_amount=200"

# Sale
curl -s -X POST -H "Content-Type: application/json" \
  -H "X-Tenant-Token: ..." -H "X-Cashier-Token: ..." \
  -d '{"client_phone":"+992900000500","total_amount":200,"bonus_redeem_amount":10}' \
  https://api.bonex.one/api/v1/cashier/sale
```

Демо-данные: `php artisan bonex:provision-demo`.