

BONEX · LOYALTY PLATFORM

Bonex — Partner API

Документация по API для интеграции

Версия от 2026-06-18 · [api.bonex.one](#)

Вonex — Partner API (интеграция стороннего приложения)

Спецификация для разработчиков, встраивающих программу лояльности Vonex в **своё** мобильное приложение (например, раздел «Бонусы» в приложении сети магазинов).

Базовый URL: `https://api.bonex.one/api/v1` (поле `base_url` в panel → Интеграции → Партнёрский API).

1. Модель интеграции

Сторона	Что делает
Вonex	Хранит клиентов, баланс, начисления/списания, акции, уровни.
Бэкенд партнёра	Знает телефон авторизованного пользователя. Подписывает server-to-server запросы (HMAC). Хранит Tenant Token и Секрет .
Приложение партнёра	Показывает раздел «Бонусы»: баланс, история, акции, QR. Работает под Bearer-токеном клиента.

Сеть = один арендатор (tenant) в Vonex. Владелец магазина включает Партнёрский API в **panel** → **Интеграции** → **Партнёрский API** и передаёт программисту три значения:

- `base_url` — `https://api.bonex.one/api/v1`
- `Tenant Token` — 64 символа, идентификатор сети
- `Секрет подписи` — 64 символа, общий секрет для HMAC

Безопасность: `Tenant Token` и `Секрет` хранятся **только на сервере партнёра**. Не зашивайте их в мобильное приложение — оттуда их легко извлечь.

2. Два уровня доступа

2.1. Server-to-server (/partner/*) — HMAC-подпись

Чувствительные операции вызывает **бэкенд партнёра**. Заголовки:

```
X-Tenant-Token: <Tenant Token>
X-Partner-Timestamp: <unix-время в секундах>
X-Partner-Signature: <HMAC-SHA256 hex>
Content-Type: application/json
Accept: application/json
```

Строка подписи:

```
{timestamp}.{METHOD}.{path}.{rawBody}
```

- `timestamp` — то же значение, что и в заголовке (Unix-секунды);
- `METHOD` — HTTP-метод в верхнем регистре (`GET` , `POST`);
- `path` — путь запроса, например `/api/v1/partner/sale` ;
- `rawBody` — тело запроса **байт-в-байт** (для `GET` — пустая строка).

Подпись: `HMAC-SHA256(signing_string, Секрет)` → hex.

Метка времени должна отличаться от времени сервера не более чем на **300 секунд** (защита от повторного воспроизведения). Синхронизируйте часы (NTP).

Пример (Node.js)

```
const crypto = require('crypto');

function signedHeaders({ tenantToken, secret, method, path, body }) {
  const ts = Math.floor(Date.now() / 1000).toString();
  const raw = body ? JSON.stringify(body) : '';
  const signing = `${ts}.${method.toUpperCase()}.${path}.${raw}`;
  const sig = crypto.createHmac('sha256', secret).update(signing).digest('hex');
  return {
    'X-Tenant-Token': tenantToken,
    'X-Partner-Timestamp': ts,
    'X-Partner-Signature': sig,
    'Content-Type': 'application/json',
  };
}
```

Тело запроса в подписи и в HTTP должно совпадать байт-в-байт. Сериализуйте JSON один раз и отправляйте ту же строку.

Пример (PHP)

```
$ts = (string) time();
$raw = json_encode($body, JSON_UNESCAPED_UNICODE);
$signing = "{$ts}.POST./api/v1/partner/sale.{$raw}";
$sig = hash_hmac('sha256', $signing, $secret);
```

2.2. Клиентский доступ (/client/*) — Bearer-токен

После `POST /partner/auth` приложение получает **Bearer-токен клиента** и обращается к read-эндпоинтам напрямую (нужен только `X-Tenant-Token` + `Authorization`):

```
X-Tenant-Token: <Tenant Token>
Authorization: Bearer <token клиента>
```

3. Эндпоинты

3.1. SSO — обмен телефона на токен клиента

POST /partner/auth

Бэкенд партнёра вызывает после входа пользователя у себя. Вonex находит клиента по телефону (или создаёт нового) и возвращает Bearer-токен для приложения.

```
{
  "phone": "+992901234567",
  "full_name": "Парвиз Солиев",
  "create_if_missing": true
}
```

Поле	Обяз.	Описание
phone	да	Телефон пользователя (нормализуется на стороне Вonex)
full_name	нет	Имя — для создания нового клиента
create_if_missing	нет	true (по умолч.) — создать клиента, если его нет; false → 404

Ответ 200/201:

```
{
  "success": true,
  "data": {
    "token": "12|abcdef ... ",
    "created": false,
    "client": {
      "id": 42,
      "phone": "+992901234567",
      "full_name": "Парвиз Солиев",
      "bonus_balance": 150,
      "barcode_ean13": "200000000042X",
      "tier": { "id": 2, "name": "Gold", "color": "#C9A227" }
    }
  }
}
```

token — Sanctum-токен клиента. Передайте его в приложение для вызова /client/* .

3.2. Чтение бонусов (приложение, Bearer-токен)

Уже существующие клиентские эндпоинты (питают Telegram Mini App и Flutter-приложение):

Метод	Путь	Назначение
GET	<code>/client/profile</code>	Баланс, доступно к списанию, уровень, QR/EAN-13
PUT	<code>/client/profile</code>	Обновить имя / дату рождения / пол
GET	<code>/client/transactions</code>	История начислений/списаний (пагинация)
GET	<code>/client/purchases</code>	История покупок
GET	<code>/client/promotions</code>	Персональные акции
GET	<code>/client/referral</code>	Реферальная программа
GET	<code>/client/settings</code>	Брендинг (название, цвет, валюта) — без авторизации, только Tenant Token
POST	<code>/client/device-token</code>	Регистрация push-токена (FCM)

Пример профиля:

```
{
  "success": true,
  "data": {
    "id": 42,
    "phone": "+992901234567",
    "full_name": "Парвиз Солиев",
    "bonus_balance": 150,
    "barcode_ean13": "2000000000420",
    "birth_date": "1990-05-01",
    "tier": { "id": 2, "name": "Gold", "color": "#C9A227" }
  }
}
```

QR для кассы: закодируйте `phone` в международном формате (`+992901234567`) либо покажите `barcode_ean13` . Касса сканирует это и привязывает клиента к чеку.

3.3. Поиск клиента и баланса (server-to-server)

```
GET /partner/client/{identifier}?check_amount=200
```

`identifier` — URL-encoded телефон (`%2B992 ...`), EAN-13 или клубная карта. `check_amount` (опц.) — сумма заказа для расчёта `max_redeem_amount` .

```
{
  "success": true,
  "data": {
    "id": 42,
    "phone": "+992901234567",
    "bonus_balance": 150,
    "redeemable_balance": 120,
    "held_balance": 30,
    "loyalty": {
      "bonus_percent": 5,
      "max_redeem_percent": 30,
      "redeem_hold_human": "24 ч",
      "redeemable_balance": 120,
      "max_redeem_amount": 60
    }
  }
}
```

- `redeemable_balance` — доступно к списанию сейчас (бонусы после выдержки 24 ч).
- `held_balance` — на выдержке, списать нельзя.
- `max_redeem_amount = min(check_amount × max_redeem_percent / 100, redeemable_balance)`.

3.4. Продажа: начисление (+ опционально списание)

```
POST /partner/sale
```

Вызывается, когда у партнёра прошла покупка/заказ. Начисляет кэшбэк и (если передан `bonus_redeem_amount`) сразу списывает бонусы.

```
{
  "client_phone": "+992901234567",
  "total_amount": 955.00,
  "gross_amount": 1000.00,
  "bonus_redeem_amount": 45.00,
  "redeem_mode": "discount",
  "external_sale_id": "order-2026-000123",
  "receipt_number": "000123",
  "sale_date": "2026-06-15T14:30:00",
  "lines": [
    { "sku": "SKU-001", "name": "Молоко 1л", "quantity": 2, "unit_price": 20, "line_amount": 40 }
  ]
}
```

Поле	Обяз.	Описание
client_phone	да	Телефон клиента
total_amount	да	Итог к оплате (после скидки бонусами в режиме discount)
gross_amount	нет	Полная сумма до списания (для discount)
bonus_redeem_amount	нет	Сколько бонусов списать
redeem_mode	нет	payment_type (по умолч.) или discount
external_sale_id	нет	Ваш ID заказа — идемпотентность (повтор → 422)
lines	нет	Строки заказа (для истории и тегов интересов)

Ответ 201:

```
{
  "success": true,
  "data": {
    "id": 1001,
    "total_amount": 955,
    "bonus_accrued": 47.75,
    "bonus_redeemed": 45,
    "new_balance": 152.75
  }
}
```

3.5. Списание бонусов в приложении (онлайн-оплата)

POST /partner/redeem

Доступно, только если владелец включил «**Разрешить списание из приложения**» (иначе 403). Для сценария «оплатить бонусами» в онлайн-заказе.

```
{
  "client_phone": "+992901234567",
  "amount": 50,
  "total_amount": 200,
  "external_sale_id": "order-2026-000124"
}
```

Ответ 200:

```
{
  "success": true,
  "data": {
    "transaction_id": 5012,
    "amount_redeemed": 50,
    "new_balance": 102.75,
    "sale_id": 1002
  }
}
```

Ошибки 422: сумма больше доступного, превышение лимита % от чека, бонусы ещё на выдержке.

Чаще достаточно одного `POST /partner/sale` с `bonus_redeem_amount` — он и спишет, и начислит.

4. Webhooks (Bonex → бэкенд партнёра)

Если в panel указан **Webhook URL**, Bonex отправляет `POST` при изменении баланса.

Заголовки:

```
X-Bonex-Event:      bonus.accrued
X-Bonex-Timestamp:  1718450000
X-Bonex-Signature:  <HMAC-SHA256 hex>
Content-Type:       application/json
```

Проверка подписи (тем же секретом):

```
signature = HMAC-SHA256("{X-Bonex-Timestamp}.{rawBody}", Секрет)
```

Тело:

```
{
  "event": "bonus.accrued",
  "occurred_at": "2026-06-15T14:30:05+05:00",
  "amount": 47.75,
  "client": {
    "id": 42,
    "phone": "+992901234567",
    "full_name": "Парвиз Солиев",
    "bonus_balance": 152.75
  },
  "data": { "transaction_id": 1234, "sale_id": 1001, "external_sale_id": "order-2026-000123" }
}
```

События: `bonus.accrued`, `bonus.redeemed`, `bonus.expired`.

Требования к приёмнику:

- Отвечайте `2xx` быстро (≤ 10 с). Иначе Bonex повторит доставку (до 5 раз с возрастающей паузой).

- Обработывайте идемпотентно (повтор того же события возможен).
- Проверяйте подпись и свежесть `X-Bonex-Timestamp`.

5. Ошибки

Код	Значение
401	Нет/неверная подпись, просроченный timestamp, нет Tenant Token
403	Партнёрский API выключен, клиент заблокирован, списание из приложения выключено
404	Клиент не найден
422	Ошибка валидации / бизнес-правила (лимит списания, выдержка, дубль <code>external_sale_id</code>)
429	Превышен лимит запросов (600/мин на арендатора)

Формат ошибки:

```
{ "success": false, "error": "Текст ошибки." }
```

6. Чеклист подключения

1. Владелец магазина: **panel** → **Интеграции** → **Партнёрский API** → **Включить**.
2. Скопировать `base_url`, `Tenant Token`, `Секрет`; (опц.) указать `Webhook URL`.
3. Бэкенд партнёра: реализовать подпись HMAC (раздел 2.1), вызвать `POST /partner/auth`.
4. Приложение: показывать `/client/profile`, `/client/transactions`, `/client/promotions`, QR.
5. (Опц.) `POST /partner/sale` при покупке; `POST /partner/redeem` для онлайн-оплаты бонусами.
6. (Опц.) Принимать webhooks и проверять подпись.

7. Тестовый сценарий (curl)

```
BASE="https://api.bonex.one/api/v1"
TENANT="<tenant_token>"
SECRET="<secret>"
TS=$(date +%s)
BODY='{"phone":"+992900000500","create_if_missing":true}'
SIG=$(printf '%s' "${TS}.POST./api/v1/partner/auth.${BODY}" | openssl dgst -sha256 -hmac "$SECRET"

curl -s -X POST "$BASE/partner/auth" \
  -H "X-Tenant-Token: $TENANT" \
  -H "X-Partner-Timestamp: $TS" \
  -H "X-Partner-Signature: $SIG" \
  -H "Content-Type: application/json" \
  -d "$BODY"
```