BONEX · LOYALTY PLATFORM

Bonex — Mobile API

Документация по API для интеграции

Версия от 2026-06-18 · api.bonex.one

Bonex — Mobile API (интеграция мобильного приложения)

Спецификация для разработчиков, которые встраивают программу лояльности Bonex в **мобильное приложение** (iOS / Android / Flutter / React Native) или Telegram Mini App.

Приложение само авторизует клиента (по SMS-коду или через Telegram), получает **Bearer-токен** и обращается к эндпоинтам `/client/*`: баланс, история, акции, реферальная программа, push.

> Если у вас **уже есть свой бэкенд** и пользователь авторизован у вас (например, в приложении сети магазинов), используйте серверный сценарий с HMAC-подписью — см. `PARTNER_API.md`. Этот документ — про приложение, которое логинит клиента напрямую через Bonex.

Базовый URL: `https://api.bonex.one/api/v1`

---

1. Что нужно от владельца магазина

Сеть/магазин = один арендатор (tenant) в Bonex. Владелец передаёт разработчику **одно** значение:

Параметр	Где взять	Назначение
`Tenant Token`	panel.bonex.one → **Интеграции** → **Учётные данные API**	64-символьный идентификатор магазина

`Tenant Token` отправляется в каждом запросе в заголовке `X-Tenant-Token`. Он идентифицирует магазин (клиентскую базу, баланс, акции, брендинг), но **не** даёт доступа к чужим данным — конкретного клиента авторизует его персональный Bearer-токен.

---

2. Заголовки и формат ответа

Заголовки

Публичные эндпоинты (авторизация, брендинг):

```
X-Tenant-Token: <Tenant Token>
Content-Type:   application/json
Accept:         application/json
```

Эндпоинты клиента (`/client/*`) — дополнительно Bearer-токен:

```
X-Tenant-Token: <Tenant Token>
Authorization:  Bearer <token клиента>
Content-Type:   application/json
Accept:         application/json
```

Единый формат ответа

Успех:

```
{ "success": true, "data": { } }
```

Ошибка:

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

HTTP-код отражает результат (`200` , `201` , `401` , `403` , `404` , `422` , `429`). Поле `success` дублирует его булевым флагом — удобно для единой обработки в приложении.

CORS разрешён для всех источников (`Access-Control-Allow-Origin: *`), поддерживаются заголовки `Authorization` , `X-Tenant-Token` , `Content-Type` , `Accept` .

---

3. Авторизация клиента

Приложение поддерживает два способа входа: **по SMS-коду (OTP)** и **через Telegram**. Оба возвращают Bearer-токен (Laravel Sanctum), который приложение хранит и подставляет в `Authorization: Bearer ...` . Токен не имеет срока истечения — действует, пока клиент не разлогинится (в этом случае удалите токен локально).

3.1. Отправить SMS-код

```
POST /auth/send-otp
```

```
{ "phone": "+992901234567" }
```

Поле	Обяз.	Описание
`phone`	да	Телефон, до 20 символов. Нормализуется на стороне Bonex

Ответ 200:

```
{ "success": true, "data": { "message": "OTP отправлен." } }
```

Код состоит из **6 цифр**, живёт **180 секунд**, допускается до **3 попыток** ввода.

3.2. Войти по коду (существующий клиент)

```
POST /auth/verify-otp
```

```
{ "phone": "+992901234567", "code": "123456" }
```

Поле	Обяз.	Описание
`phone`	да	Телефон
`code`	да	Ровно 6 цифр из SMS

Ответ 200:

```
{
  "success": true,
  "data": {
    "token": "12|aBcD … plainTextToken",
    "client_id": 42
  }
}
```

Ошибки:

- `422` — `Неверный или просроченный код.`
- `404` — `Клиент не найден в программе лояльности.` (нужно зарегистрировать — см. 3.3)

> Сценарий приложения: если `verify-otp` вернул `404`, покажите экран регистрации (имя) и вызовите `POST /auth/register` с тем же `code`, пока он не истёк.

3.3. Регистрация нового клиента

```
POST /auth/register
```

```
{
  "phone": "+992901234567",
  "code": "123456",
  "full_name": "Парвиз Солиев",
  "referral_code": "AB12CD34"
}
```

Поле	Обяз.	Описание
`phone`	да	Телефон
`code`	да	Код из SMS (тот же, что для verify)
`full_name`	нет	Имя клиента
`referral_code`	нет	Реферальный код пригласившего (8 символов или вид `ref_AB12CD34`)

Ответ 201:

```
{
  "success": true,
  "data": {
    "token": "13|eFgH ... plainTextToken",
    "client_id": 43
  }
}
```

Ошибки:

- `422` — `Неверный или просроченный код.`
- `422` — `Клиент уже зарегистрирован. Войдите по коду.`
- `422` — ошибки бизнес-правил (неверный реферальный код, превышен лимит тарифа магазина).

3.4. Вход через Telegram (Mini App)

Для Telegram Web App: приложение получает `initData` из Telegram SDK и передаёт её в Bonex, который проверяет подпись бота магазина.

```
POST /auth/telegram
```

```
{ "init_data": "<window.Telegram.WebApp.initData>" }
```

Ответ 200 — клиент уже привязан к Telegram:

```
{
  "success": true,
  "data": {
    "token": "14|iJkL ... plainTextToken",
    "client": { "id": 42, "phone": "+992901234567", "full_name": "Парвиз Солиев", "bonus_balance":
  }
}
```

`client` — полный объект клиента (см. раздел 5.1).

Ошибки:

- `400` — `Telegram-бот не настроен для магазина.`
- `401` — `Недействительная подпись Telegram.`
- `404` — `Аккаунт не привязан. Поделитесь контактом для входа.` → перейдите к 3.5.

3.5. Первый вход через Telegram (привязка по контакту)

Если по `initData` клиент не найден, приложение запрашивает контакт у пользователя (`requestContact`) и отправляет его вместе с `initData`. Bonex найдёт клиента по телефону или создаст нового и привяжет Telegram.

```
POST /auth/telegram-contact
```

```
{
  "init_data": "<initData>",
  "contact": "<подписанный контакт из Telegram>",
  "referral_code": "AB12CD34"
}
```

Поле	Обяз.	Описание
`init_data`	да	`initData` Telegram Web App
`contact`	да	Объект контакта (с подписью), полученный из Telegram
`referral_code`	нет	Реферальный код пригласившего

Ответ 200/201 — как в 3.4 (`token` + `client`). Код `201` , если клиент был создан.

Ошибки: `401` (подпись), `422` (контакт не совпадает с аккаунтом / Telegram уже привязан к другому клиенту), `403` (клиент заблокирован).

---

4. Брендинг магазина (без авторизации)

Для splash-экрана и темизации приложения до входа клиента. Нужен только `X-Tenant-Token` .

```
GET /client/settings
```

Ответ 200:

```
{
  "success": true,
  "data": {
    "store_name": "Сеть «Хорошо»",
    "logo_url": "https://cdn.bonex.one/logos/42.png",
    "primary_color": "#0E7C66",
    "currency": "TJS",
    "bonus_unit": "бонус"
  }
}
```

> Хотя путь начинается с `/client` , авторизация для `GET /client/settings` не требуется — токен клиента не нужен, только `X-Tenant-Token` .

---

5. Эндпоинты клиента (Bearer-токен)

Все вызовы ниже требуют `X-Tenant-Token` + `Authorization: Bearer <token>` . Лимит: **60 запросов/мин** на клиента (см. раздел 8).

5.1. Профиль и баланс

```
GET /client/profile
```

Ответ 200:


```
{
  "success": true,
  "data": {
    "id": 42,
    "phone": "+992901234567",
    "barcode_ean13": "2000000000420",
    "club_card_number": "001-0000123",
    "account_number": "ACC-000042",
    "full_name": "Парвиз Солиев",
    "birth_date": "1990-05-01",
    "gender": "male",
    "bonus_balance": 150.0,
    "welcome_bonus_balance": 0.0,
    "welcome_expires_at": null,
    "birthday_bonus_balance": 0.0,
    "birthday_expires_at": null,
    "total_spent": 1200.0,
    "tier": { "id": 2, "name": "Gold", "color": "#C9A227" },
    "telegram_username": "parviz",
    "has_mobile_app": true,
    "is_blocked": false,
    "last_visit_at": "2026-06-10T12:30:00+00:00"
  }
}
```

Поле	Тип	Описание
`id`	int	ID клиента в Bonex
`phone`	string	Телефон в международном формате
`barcode_ean13`	string	Штрих-код карты (EAN-13) для кассы
`club_card_number`	string	Номер клубной карты
`account_number`	string	Лицевой счёт
`full_name`	string	Имя
`birth_date`	date	Дата рождения (`YYYY-MM-DD`) или `null`
`gender`	string	Пол или `null`
`bonus_balance`	float	Общий баланс бонусов
`welcome_bonus_balance`	float	Приветственные бонусы (отдельный кошелёк)
`welcome_expires_at`	datetime	Когда сгорят приветственные бонусы (ISO 8601) или `null`
`birthday_bonus_balance`	float	Бонусы ко дню рождения
`birthday_expires_at`	datetime	Срок их сгорания или `null`
`total_spent`	float	Сумма покупок за всё время
`tier`	object	Уровень лояльности: `id` , `name` , `color`
`telegram_username`	string	Telegram, если привязан, иначе `null`
`has_mobile_app`	bool	Зарегистрирован ли push-токен
`is_blocked`	bool	Заблокирован ли клиент
`last_visit_at`	datetime	Последний визит (ISO 8601) или `null`

5.2. Обновить профиль

```
PUT /client/profile
```

```
{ "full_name": "Парвиз Солиев", "birth_date": "1990-05-01", "gender": "male" }
```

Поле	Обяз.	Описание
`full_name`	нет	Имя, до 255 символов
`birth_date`	нет	Дата (`YYYY-MM-DD`)
`gender`	нет	Строка, до 20 символов

Ответ 200: обновлённый объект клиента (как в 5.1).

5.3. История бонусов (начисления/списания)

```
GET /client/transactions
```

Пагинация: по 20 записей на страницу. Параметр `?page=` (стандартный Laravel).

Ответ 200:


```
{
  "success": true,
  "data": {
    "data": [
      {
        "id": 5012,
        "type": "accrual",
        "amount": 7.5,
        "amount_unit": "бонус",
        "balance_after": 157.5,
        "sale_amount": 1000.0,
        "purchase_currency": "TJS",
        "note": "Начисление за покупку",
        "created_at": "2026-06-10T12:30:05+00:00"
      }
    ],
    "current_page": 1,
    "last_page": 4,
    "total": 73
  }
}
```


`type` — один из: `accrual` (начисление), `redeem` (списание), `welcome`, `birthday`, `referral_inviter`, `referral_invitee`, `expire` и др. `amount` положительный для начислений, отрицательный для списаний/сгораний. `sale_amount` / `purchase_currency` присутствуют, если транзакция связана с покупкой.

5.4. История покупок

```
GET /client/purchases?per_page=20
```

`per_page` — от 1 до 50 (по умолчанию 20). Пагинация через `?page=`.

Ответ 200:

```
{
  "success": true,
  "data": {
    "data": [
      {
        "id": 1001,
        "kind": "purchase",
        "title": "Покупка",
        "receipt_number": "000123",
        "sale_date": "2026-06-10T12:30:00+00:00",
        "total_amount": 1000.0,
        "currency": "TJS",
        "bonus_accrued": 7.5,
        "bonus_redeemed": 45.0,
        "store_name": "Магазин на Рудаки",
        "store_address": "пр. Рудаки, 12",
        "lines": [
          { "sku": "SKU-001", "name": "Молоко 1л", "quantity": 2.0, "unit_price": 20.0, "line_amou
        ]
      }
    ],
    "current_page": 1,
    "last_page": 2,
    "total": 31
  }
}
```


`lines` — товарные позиции чека (может быть пустым, если касса их не передавала).

5.5. Акции

```
GET /client/promotions
```

Для авторизованного клиента возвращаются **персональные** акции (с учётом его сегмента):


```
{
  "success": true,
  "data": [
    {
      "id": 7,
      "name": "Двойной кэшбэк на молочное",
      "description": "x2 бонусов на категорию DAIRY",
      "type": "category_multiplier",
      "condition": { "category_code": "DAIRY" },
      "reward": { "multiplier": 2 },
      "starts_at": "2026-06-01T00:00:00+00:00",
      "ends_at": "2026-06-30T23:59:59+00:00",
      "personalized": true
    }
  ]
}
```

Если запрос сделан без Bearer-токена (только `X-Tenant-Token`), вернётся публичный список активных акций без персональных (поля `id`, `name`, `description`, `type`, `starts_at`, `ends_at`).

5.6. Список магазинов/филиалов

```
GET /client/stores
```

```
{
  "success": true,
  "data": [
    { "id": 1, "name": "Магазин на Рудаки", "address": "пр. Рудаки, 12" }
  ]
}
```

5.7. Реферальная программа

```
GET /client/referral
```

```
{
  "success": true,
  "data": {
    "code": "AB12CD34",
    "share_text": "Пригласи друга в Bonex — получи 20 бонусов. Друг получит 10 бонусов после перво
    "telegram_start_param": "ref_AB12CD34",
    "invite_url": "https://t.me/yourbot?start=ref_AB12CD34",
    "invited_count": 5,
    "completed_count": 3,
    "bonuses_earned": 60.0
  }
}
```

Поле	Описание
`code`	Реферальный код клиента (генерируется автоматически)
`share_text`	Готовый текст для кнопки «Поделиться»
`telegram_start_param`	Параметр `start` для deep-link Telegram
`invite_url`	Ссылка-приглашение (или `null`, если бот не настроен)
`invited_count`	Сколько друзей зарегистрировалось по коду
`completed_count`	Сколько из них совершили первую покупку (награда выдана)
`bonuses_earned`	Сколько бонусов клиент заработал на рефералах

5.8. Привязать Telegram к текущему аккаунту

```
POST /client/link-telegram
```

```
{ "init_data": "<initData Telegram Web App>" }
```

Ответ 200: `{ "success": true, "data": { "linked": true } }`

Ошибки: `400` (бот не настроен), `401` (подпись), `422` (Telegram уже привязан к другому клиенту).

---

6. Push-уведомления (FCM)

Чтобы клиент получал push при начислении/списании бонусов и напоминания о сгорании, зарегистрируйте FCM-токен устройства после входа.

Регистрация токена

```
POST /client/device-token
```

```
{ "token": "<FCM registration token>", "platform": "android" }
```

Поле	Обяз.	Описание
`token`	да	FCM-токен устройства, до 512 символов
`platform`	нет	`android` или `ios`

Ответ 200: `{ "success": true, "data": { "registered": true } }`

Токен уникален: при повторной регистрации он переназначается текущему клиенту.

Удаление токена (logout / отключение push)

```
DELETE /client/device-token
```

```
{ "token": "<FCM registration token>" }
```

Ответ 200: `{ "success": true, "data": { "deleted": true } }`

> Bonex шлёт уведомления асинхронно: **Push** (если есть device token) → иначе **Telegram** (если привязан) → иначе **SMS** (если магазин подключил SMS). Само приложение уведомления не отправляет.

---

7. QR / карта клиента для кассы

Чтобы кассир начислил бонусы, приложение показывает идентификатор клиента. Касса (1С, Frontol, веб-касса) сканирует одно из:

- **QR-код** — закодируйте `phone` в международном формате: `+992901234567`;

- **штрих-код** — выведите `barcode_ean13` как EAN-13;
- либо клиент называет телефон голосом.

Отдельный «зашифрованный токен» не нужен — идентификатор клиента в Bonex = нормализованный телефон. Подробнее о кассовой стороне — `INTEGRATIONS_API.md`.

---

8. Лимиты запросов

Группа	Лимит	Ключ
Авторизованный клиент (`/client/*`)	**60 запросов/мин**	по клиенту (или IP)

При превышении — `429 Too Many Requests`. Делайте экспоненциальную паузу и не опрашивайте профиль в цикле — кэшируйте баланс и обновляйте по событию (pull-to-refresh, push).

---

9. Коды ошибок

Код	Значение
`200` / `201`	Успех (201 — создан новый ресурс/клиент)
`401`	Нет/неверный `X-Tenant-Token`, неверная подпись Telegram, или истёк/неверен Bearer-токен
`403`	Магазин приостановлен / подписка не оплачена, либо клиент заблокирован
`404`	Клиент не найден / не зарегистрирован
`422`	Ошибка валидации или бизнес-правила (неверный код, дубликат, лимит тарифа)
`429`	Превышен лимит запросов

Тело ошибки всегда: `{ "success": false, "error": "..." }`.

Частые сообщения:

- `X-Tenant-Token header is required.` — не передан токен магазина.
- `Недействительный токен, подписка не оплачена или магазин приостановлен.` — проверьте Tenant Token / статус магазина.
- `Неверный или просроченный код.` — OTP истёк (180 с) или введён неверно.
- `Клиент не найден в программе лояльности.` — нужно зарегистрировать (`/auth/register`).

---

10. Типовой сценарий приложения

1. При запуске: `GET /client/settings` → тема, логотип, валюта (без токена).
2. Экран входа: `POST /auth/send-otp` → `POST /auth/verify-otp`.
   - Если `404` — экран регистрации → `POST /auth/register`.
   - Сохранить `token` в защищённое хранилище (Keychain / Keystore).

3. После входа: `POST /client/device-token` (FCM).
4. Главный экран: `GET /client/profile` → баланс, уровень, QR/EAN-13.
5. Вкладки: `GET /client/transactions`, `GET /client/purchases`, `GET /client/promotions`, `GET /client/referral`.
6. Экран оплаты: показать QR с телефоном клиента для кассы.
7. Logout: `DELETE /client/device-token` + удалить локальный токен.

---

11. Быстрый тест (curl)

```
BASE="https://api.bonex.one/api/v1"
TENANT="<tenant_token>"

# 1. Отправить код
curl -s -X POST "$BASE/auth/send-otp" \
  -H "X-Tenant-Token: $TENANT" -H "Content-Type: application/json" \
  -d '{"phone":"+992900000500"}'

# 2. Войти по коду → получить token
curl -s -X POST "$BASE/auth/verify-otp" \
  -H "X-Tenant-Token: $TENANT" -H "Content-Type: application/json" \
  -d '{"phone":"+992900000500","code":"123456"}'

# 3. Профиль (подставьте token из шага 2)
curl -s "$BASE/client/profile" \
  -H "X-Tenant-Token: $TENANT" \
  -H "Authorization: Bearer 12|aBcD…"
```

---

12. Сводка эндпоинтов

Метод	Путь	Авторизация	Назначение
POST	`/auth/send-otp`	Tenant	Отправить SMS-код
POST	`/auth/verify-otp`	Tenant	Вход по коду (получить token)
POST	`/auth/register`	Tenant	Регистрация нового клиента
POST	`/auth/telegram`	Tenant	Вход через Telegram
POST	`/auth/telegram-contact`	Tenant	Первый вход через Telegram (по контакту)
GET	`/client/settings`	Tenant	Брендинг магазина
GET	`/client/profile`	Tenant + Bearer	Профиль и баланс
PUT	`/client/profile`	Tenant + Bearer	Обновить профиль
GET	`/client/transactions`	Tenant + Bearer	История бонусов
GET	`/client/purchases`	Tenant + Bearer	История покупок
GET	`/client/promotions`	Tenant + Bearer	Персональные акции
GET	`/client/stores`	Tenant + Bearer	Список магазинов
GET	`/client/referral`	Tenant + Bearer	Реферальная программа
POST	`/client/link-telegram`	Tenant + Bearer	Привязать Telegram
POST	`/client/device-token`	Tenant + Bearer	Зарегистрировать push-токен
DELETE	`/client/device-token`	Tenant + Bearer	Удалить push-токен

13. Связанные документы

Документ	Когда использовать
`MOBILE_API.md` (этот файл)	Приложение логинит клиента напрямую через Bonex (OTP / Telegram)
`PARTNER_API.md`	У партнёра есть свой бэкенд: server-to-server SSO с HMAC-подписью, продажи, списание, webhooks
`INTEGRATIONS_API.md`	Кассовое ПО: 1С (УНФ/Розница/УТ), Frontol, веб-касса