

Личный кабинет Юмобайл: web-app-umobile

Для разработчиков. Состояние на 28.09.2026. Переменные, ручки и выкатка подробно — в README репозитория `Buharestskie/web-app-umobile`; карта «экран → запрос → поля» — `API-MAP.md` там же.

Коротко

Личный кабинет абонента Юмобайла (MVNO на сети билайн): баланс, тариф, остатки, статус номера, несколько номеров на одном аккаунте, пополнение через СБП и картой, новости, чат с поддержкой, уведомления и активация новой SIM — от скана серийника до подписания договора в Госключе или проверки паспорта. Данные абонента — не наши: их отдаёт API оператора, приложение показывает ровно то, что он вернул. Своё у нас — два бэкенда: чат и уведомления, и выдача номеров при активации.

Часть	Папка	Что это	Живое?
Веб-кабинет (PWA)	<code>web/</code>	React 19 + TypeScript + Vite, открывается на <code>app.umobile.ru/app/</code> , ставится на экран «Домой»	да, основная форма продукта
Чат и уведомления	<code>chat-backend/</code>	FastAPI + Postgres: чат с оператором через amoCRM, лента уведомлений, web-push и FCM, ведение сделки активации в amoCRM	да
Сервис активации	<code>activation-admin/</code>	FastAPI + Postgres: пул номеров и партии SIM, выдача номера при скане, привязка в ЛК оператора, сессии, админка	да
Нативная оболочка iOS	<code>native/</code>	SwiftUI + WKWebView; нативные чат, сканер SIM, пуши через Firebase	заморожена с 26.08.2026, см. ниже

Часть	Папка	Что это	Живое?
Конфиги nginx	server/	справочные копии боевых конфигов обоих доменов и зоны микрокэша	справочно
Выкатка	deploy/ , .github/workflows/ci.yml	скрипт выкатки и CI	да
Документация	корень	lk_umobile_api.yaml (спека API оператора), API-MAP.md, SPEC-*.md и TZ-*.md — задачи для разработчика API оператора, ПИСЬМО-АВТОРУ-API*.md	справочно

AGENT_CONTEXT.md местами описывает прежний фронт (оболочка app.html с экранами в iframe). Его убрали из дерева, он лежит под тегом legacy-frontend. Всё про экраны сверяйте с web/src.

Откуда данные и куда уходят

- **API оператора (ЛК)** — lk.umobile.ru/api/v1, не наш. Отдаёт вход по коду из SMS, регистрацию нового абонента, профиль, данные номера (баланс, статус, тариф, остатки), новости, регистрацию в Госключе и её статус, проверку паспорта. Из браузера приложение ходит туда через наш nginx по пути /lkapi/. Спецификация — lk_umobile_api.yaml в корне репозитория.
- **Оплата** — отдельный домен оператора umobile.ru, через наш путь /payapi/: создание заказа (СБП или карта), статус, чек. Список банков для СБП приложение берёт напрямую у НСПК (qr.nspk.ru) — это единственный запрос мимо нашего прокси.
- **Чат-бэкенд** (/api/v1, /ws/) — сообщения абонента уходят в чат amoCRM (канал amojo), ответы оператора приходят вебхуком и доставляются в открытое приложение по WebSocket, а в закрытое — пушем. Сюда же биллинг оператора шлёт события о пополнении и смене статуса SIM (POST /api/v1/notify, типы balance, status, info).
- **Сервис активации** (/actapp/) — выдаёт номер из своего пула, когда абонент сканирует пустышку, и привязывает его в ЛК оператора вместе с серийником SIM и id сделки amoCRM.

С какими нашими сервисами связан

- **amoCRM.** Чат-бэкенд ходит в amoCRM сам, напрямую, со своим ограничителем частоты — не через шлюз `amo-queue.donula26.ru`. Аккаунт задаётся в `AMO_SUBDOMAIN`, это не основной аккаунт donula. Воронка «Активация 2026». ID воронки, этапов и полей («Выбрал номер», «Серийный номер SIM», «Ссылка на ЛК») — константы в `chat-backend/funnel.py`.
- **Мониторинг.** У чата и активации есть внешние пробы, но на публичной статус-странице они скрыты.
- Других сервисов на `*.donula26.ru` код не вызывает.

Как устроено

Веб-кабинет

- Маршруты — в адресной строке (`web/src/app/App.tsx`). Вкладки: «Главная», «Новости», «Поддержка» (чат), «Профиль». Отдельно: вход и восстановление пароля, заказ SIM (ссылки на магазины Ozon и Wildberries), итог оплаты `/pay/:result`, установка на «Домой» `/install`, вход по ссылке `/magic`, флоу активации `/activation/*`.
- Все адреса API относительные (`web/src/shared/config.ts`). Всё проксирует nginx того же домена, прямые адреса оператора из браузера не используются.
- Главная одна на все состояния номера. Подписи статусов — в `web/src/shared/lib/numberStatus.ts`: «Активен», «Блокирован», «Ожидает оплаты», «Ожидает регистрацию», «Активируется». Незнакомый статус подписывается нейтрально.
- Service worker (`web/src/sw.ts`) отвечает только за пуши и клик по уведомлению, офлайн-кеша нет. HTML отдаётся без кэша, ассеты — с хешем в имени.
- Иконочный шрифт хранится у нас (`public/fonts/ms.woff2`): Google Fonts из РФ недоступен.

Активация SIM

1. Контактный номер → код из SMS. Код шлёт и проверяет ЛК оператора (регистрация по коду), в amoCRM заводится сделка.
2. Скан серийника пустышки. Сервис активации бронирует номер из пула на 30 минут, берёт у чат-бэкенда id сделки, привязывает номер в ЛК. Пул разделён на B2C и B2B, пропорция выдачи настраивается в админке.
3. Оплата, дальше развилка по сегменту номера: B2C подписывает договор через Госуслуги и Госключ, B2B снимает паспорт своей камерой с рамкой (`/activation/kyc`).
4. На каждом шаге фронт двигает сделку по этапам воронки (`POST /api/v1/activation/stage`).

Статусы в базе активации. Номер: `pool` (свободен), `res` (бронь), `activ` (активируется), `act` (активен), `taken` (уже занят в ЛК), `blk`, `quar` (карантин). SIM: `ship` → `pay` → `prov` → `bound`, отдельно `blk`. Каждое движение номера пишется в журнал `number_log`, в админке он открывается щелчком по номеру или ICCID. Сама таблица `numbers` хранит только текущее состояние.

Нативная оболочка

Приложение `native/` грузит не React-сборку, а свою офлайн-копию прежнего фронта (`native/Yumobile/app.html` и `screens/`). Чат, регистрацию пуш-токена и часть API берёт с `app.umobile26.ru`. Собирается скриптом `native/build-install.sh` с бесплатной подписью разработчика. Правки в `web/` в нативку не попадают.

Авторизация в общих словах

- **Абонент ↔ ЛК оператора:** вход по номеру и коду из SMS, в ответ Bearer-токен ЛК. Он хранится в браузере и идёт в `/lkapi/`.
- **Абонент ↔ наши бэкенды:** токен ЛК обменивается на токен нашей сессии (`POST /actapp/session`). Сервис активации проверяет токен запросом профиля в ЛК и берёт номера абонента оттуда. Сессия живёт 30 дней и подписана `SESSION_SECRET` — он общий у обоих бэкендов, поэтому сессию, выданную активацией, принимает и чат. Ручки чата, уведомлений, подписки на пуши, скана и состояния активации сверяют номер из запроса с номерами в сессии.
- **Сервис ↔ сервис:** биллинг — заголовок со своим секретом (`NOTIFY_SECRET`), к ЛК — внутренний ключ (`LK_INTERNAL_KEY`), активация к чату — `APP_SECRET` чата.
- **Админка активации** — `app.umobile26.ru/activation/`, за basic-auth и ключом `ADMIN_KEY`.

Переменные окружения

Значения — в `.env` рядом с `docker-compose.yml` каждого бэкенда, в git их нет. Полный список чата — `chat-backend/config.py`.

Сервис	Переменные
--------	------------

chat-backend	DATABASE_URL, AMO_SUBDOMAIN, AMO_CLIENT_ID, AMO_CLIENT_SECRET, AMO_REDIRECT_URI, AMOJO_CHANNEL_ID, AMOJO_CHANNEL_CODE, AMOJO_CHANNEL_SECRET, APP_SECRET, SESSION_SECRET, NOTIFY_SECRET, LK_INTERNAL_KEY, MAGIC_SECRET, MAGIC_WORKER, VAPID_PRIVATE_PATH, VAPID_SUBJECT, FCM_PROJECT_ID, FCM_CREDENTIALS_PATH, PUBLIC_BASE_URL, WORKERS
activation-admin	DATABASE_URL, ADMIN_KEY, APP_KEY, SESSION_SECRET, LK_BASE, LK_INTERNAL_KEY, CHAT_BASE, APP_SECRET, OTP_TEST_MODE
web (при сборке)	VITE_APP_SECRET, VITE_APP_KEY, VITE_VAPID_PUBLIC_KEY — необязательные, без них берутся значения из web/src/shared/config.ts

Ссылка с автоходом

В чат-бэкенде есть догон тех, кто бросил активацию на середине. На вебхук amoCRM о новой сделке бэкенд выпускает короткую ссылку `app.umobile.ru/i/<код>` и пишет её в поле сделки «Ссылка на ЛК». По ссылке абонент попадает в кабинет уже вошедшим, без ввода номера и кода, а главная сама показывает следующий шаг. Код гасит не переход по ссылке, а запрос из приложения — так превью-боты мессенджеров не сжигают ссылку раньше человека. Фоновая рассылка напоминаний включается переменной `MAGIC_WORKER`, по умолчанию выключена. По `SPEC-magic-link.md` 03.09.2026 рассылку приостановили до ответа автора API оператора. Что включено сейчас, смотреть в `.env` на сервере.

Локальный запуск

- **Фронт:** `cd web && npm install && npm run dev`, открыть `http://localhost:5173/app/`. Дев-сервер проксирует API на боевой `app.umobile.ru`: тестового контура у оператора нет, данные настоящие.
- **Мок-режим** — чтобы открыть экраны за авторизацией без живого абонента. В одном терминале `npm run mock-api`, в другом `npm run dev:mock`. Нужный экран открывается адресом `/app/__dev?to=/app/news`. Состояние номера в моке переключается адресами мок-сервера `/__state/...`, `/__seg/...`, `/__kyc/...` (см. `web/mock-server.mjs`).
- **Бэкенды:** `docker compose up -d --build` в `chat-backend/` и `activation-admin/`. Нужны Postgres и заполненный `.env`. Чат при первом старте связывается с amoCRM через `/amo/auth`, проверка — `GET /health` (`amo_authorized`, `amojo_connected`). Админка активации — `http://localhost:8020/?k=<ADMIN_KEY>`.
- **iOS:** `cd native && ./build-install.sh sim` (нужны macOS и Xcode).

Выкатка

- **Выкатывает CI.** Пуш в `main` запускает `.github/workflows/ci.yml`: `lint → sast → build → deploy`. Ветка `dev` и pull request проходят первые три стадии без выкатки. Deploy идёт на своём раннере на боевом сервере, ключей от сервера в GitHub нет.
- Раннер запускает только `deploy/umobile-deploy.sh` с целью `web`, `activation` или `chat`. Порядок: фронт, активация, чат. Для бэкендов скрипт снимает копию текущей версии, пересобирает образ и возвращает старую версию, если новая не ответила или в контейнере активации пустые ключи.
- У чата и активации есть внешние пробы, поэтому выкатка — в окне обслуживания. Окно открывать до мержа в `main`: выкатка начинается сама.
- Конфиги nginx в `server/` — только справочные копии, CI их не выкатывает. Правка nginx делается на сервере руками.

На что наступают

- **Разработка идёт на боевых данных.** `npm run dev` ходит в боевой ЛК и боевые бэкенды. WebSocket дев-сервер проксирует на боевой домен даже в мок-режиме.
- **Мерж в `main` — это выкатка на прод**, отдельной кнопки нет.
- **Выкатка фронта заменяет каталог сайта целиком** (`rsync --delete`). Всё, чего нет в сборке `web/dist`, с сайта пропадает. Класть файлы руками рядом со сборкой бесполезно — их сотрёт следующая выкатка.
- **Бэкенды — только с пересборкой образа.** Код копируется в образ, поэтому `docker compose restart` правку не подхватит, нужен `up -d --build`.
- **Секрет со знаком `$` в `environment:` compose приезжает обрезанным.** Сервис поднимается и молча отвечает 401. Секреты передавать только через `env_file`: в `activation-admin/docker-compose.yml` `APP_SECRET` нарочно не выписан.
- **`SESSION_SECRET` должен совпадать у обоих бэкендов, а `APP_SECRET` активации — с `APP_SECRET` чата.** Разошёлся первый — все персональные ручки чата отвечают 403. Нет второго — активация не получает сделку, и каждый скан SIM отвечает `amo_lead_missing`.
- **Активация зависит от amoCRM и чат-бэкенда.** Без сделки номер в ЛК не привязывается: скан возвращает `amo_lead_missing`, бронь откатывается, номер возвращается в пул.
- **Ошибки этапов воронки не видны абоненту.** Фронт отправляет этап и игнорирует ответ, так что сбой видно только в логах чат-бэкенда.
- **Статус «Активируется» ЛК сам не отдаёт.** Приложение ставит его себе, когда видит подпись в Госключе или принятые документы.
- **ЛК виснет на частых запросах.** Поэтому `/lkapi/profile/` и `/lkapi/number/data/` идут через микрокэш nginx на 5 секунд с токеном в ключе. Состояние кэша видно в заголовке `X-LK-Cache` (`HIT`, `MISS`, `STALE`). Повтор кода входа раньше чем через минуту ЛК отклоняет с `425`.

- **Таймауты фронта:** обычный запрос — 15 секунд (`REQUEST_TIMEOUT_MS`), отправка снимков паспорта — 90 секунд (`UPLOAD_TIMEOUT_MS`). Новый запрос с тяжёлым телом на общем таймауте будет сдаваться на мобильной сети.
- **Не писать пары «свойство + `-webkit-` -двойник» руками.** Префиксы проставляет сборка по `build.cssTarget` в `web/vite.config.ts`. Иначе минификатор выкинет одну из форм, и размытие пропадёт в Chrome или Safari. В деве CSS не минифицируется, поэтому такие вещи проверять через `npm run preview`.
- **Номер, снятый с пустышки после привязки в ЛК, уходит в карантин (`quar`), а не в пул.** Метода отвязки у ЛК нет: привязку снимают в ЛК руками, потом номер переводят в «Свободен» в админке. Карантин B2C и B2B не смешивается.
- **ОТР-ручки сервиса активации (`/actapp/send-code`, `/actapp/verify-code`) SMS не отправляют,** SMS-шлюза нет. Веб подтверждает номер через ЛК оператора и эти ручки не вызывает.
- **Нативка живёт своей копией фронта.** Изменения веба туда не доезжают. Бесплатная подпись разработчика действует 7 дней, потом приложение на телефоне перестает запускаться — пересобрать `./build-install.sh device`.
- **Postgres стоит на хосте, не в контейнере.** Контейнеры ходят к нему через `host.docker.internal`. Если оба сервиса падают на старте с ошибкой подключения к базе, сначала смотреть, на каких адресах слушает Postgres.

Версия #1

Admin создал 2026-09-28 08:26:14 UTC

Admin обновил 2026-09-28 08:26:14 UTC