

WebRTC-телефония: pbx

Для разработчиков. Состояние на 28.09.2026. Подробности, полный список ручек и переменных — README репозитория `Buharestskie/pbx`. Кода самого виджета amoCRM в репозитории нет — только его серверная часть.

Коротко

pbx — собственная WebRTC-телефония отдела продаж на Asterisk. Главное в ней — автодозвонщик: менеджер ставит статус «Готов», система берёт из очереди следующий лид, звонит сначала менеджеру, после его ответа — клиенту, и соединяет их. Менеджер разговаривает из браузера: виджет в amoCRM или приложение-софтфон. После звонка в сделке появляется примечание со ссылкой на запись.

Лиды попадают в очередь из amoCRM вебхуками этапов: сделка перешла на этап — контакт встал в кампанию. Руководители настраивают кампании, операторов и смотрят дашборды в админке.

Из чего состоит

Каталог	Что это
<code>backend/</code>	Один процесс на Node.js 20 (ES-модули): HTTP API на Express (порт <code>API_PORT</code> , 3000), WebSocket для виджетов (<code>WS_PORT</code> , 3001), клиент Asterisk ARI (<code>ari-client</code>), свой клиент AMI (только для стерео-записи), PostgreSQL (<code>pg</code>). Там же фоновые задачи: ready-тик раз в 10 с, воркер вебхуков, сторожа зависших контактов и каналов, перевод из затянувшейся постобработки в «Паузу», отправка статусов менеджеров наружу.
<code>admin/</code>	Админка — статические HTML-страницы: кампании, операторы, группы, транки, график работы, история звонков, дашборды. Ходят в <code>/api/admin/...</code> по относительному пути. Вкладка «Трансфер» — пока статичный макет без запросов к API.
<code>leadboard/</code>	Одна страница: таймлайн работы с лидами МПП-2 (<code>/api/admin/dashboards/lead-timeline</code>).

Каталог	Что это
<code>guide/</code>	Инструкции для операторов и заметки о релизах виджета. Чистая статика, к API не обращается.
<code>scripts/</code>	Серверные скрипты по cron: бэкап (<code>pbx-backup.sh</code>), разбор громкости свежих записей (<code>analyze-recordings.sh</code>), генератор отчёта о постобработках (<code>regen-wrapup-report.py</code>), его же запускает кнопка в админке).

Статику backend не раздаёт: в Express её нет, страницы отдаёт веб-сервер перед ним. Скрипты в корне `backend/` (`checkleads*.mjs`, `chat-*.js`, `reconcile.mjs` и подобные) — разовые сверки, в работе сервиса не участвуют.

Чего в репозитории нет: исходников виджета amoCRM, софтбокса для Mac/Windows и расширения браузера, конфигов Asterisk (pjsip, диалплан, звуки опроса) и coturn. Бэкенд только генерирует из базы SIP-аккаунты операторов и клиентов приложения и перезагружает pjsip.

Путь звонка

- Лид в очереди.** amoCRM шлёт вебхук этапа на `/api/admin/webhook/<токен кампании>`. Ручка только кладёт id сделки в таблицу `webhook_queue`. Воркер раз в 2 с берёт до 5 строк, тянет сделку и телефон контакта из amoCRM и добавляет контакт в кампанию.
- Менеджер «Готов».** Виджет держит WebSocket с бэкендом (`auth`, `agent_status`, `sip_registered`...) и SIP-регистрацию в Asterisk через WebRTC. Звонить начинают только после `sip_registered`.
- Выбор контакта.** PL/pgSQL-функция `pick_next_contact` атомарно выдаёт контакт (`FOR UPDATE SKIP LOCKED`) с учётом приоритета кампаний, закрепления за менеджером, окна звонков по часовому поясу клиента и циклов повторов.
- Звонок менеджеру.** Виджету уходит `screen_pop` с карточкой лида, через 300 мс — ARI originate на `PJSIP/<extension>` в Stasis-приложение (`STASIS_APP`, по умолчанию `pbx-dialer`) с аргументами `agent,<callId>`.
- Менеджер ответил** (событие `StasisStart` с ролью `agent`) — виджету `dialing_client`, статус менеджера становится `oncall`, менеджер назначается ответственным за сделку в amoCRM (кроме нескольких воронок-исключений в коде). Затем originate клиенту через транк: по умолчанию шаблон `OUTBOUND_ENDPOINT_TEMPLATE`, у оператора может быть свой транк.
- Клиент ответил** (`StasisStart` с ролью `client`) — оба канала в mixing-bridge, виджету `connected`, стартует запись: стерео через AMI MixMonitor, при сбое — моно через ARI.
- Конец разговора** (`StasisEnd`) — строка в `call_log`, примечание-звонок в сделку amoCRM со ссылкой на запись, виджету `call_ended`; если клиент ответил, менеджер уходит в постобработку. Если трубку первым положил менеджер, а разговор длился

не меньше `RATING_SURVEY_MIN_TALK_SEC` (40 с), клиента оставляют на линии и просят оценить разговор цифрой 1-5 (включается `RATING_SURVEY_ENABLED=1`, оценки — в `call_ratings`).

Если SIP менеджера отвалился посреди звонка, есть восстановление: 15 с на переподключение и повторный originate. Состояние активных звонков живёт только в памяти процесса.

Другие пути через тот же диалер:

- **Ручной звонок** — `manual_call` по WebSocket или click-to-call из amoCRM (`/api/app/click-call`). Личный транк менеджера (линия «Личная») используется только здесь, автообзвон всегда идёт с общей линии.
- **Перевод МПП-1 → МПП-2** (`/api/transfer/initiate`): клоузер выбирается среди «Готовых» по rr-credit из mpp2-touches, ему уходит `screen_pop` и originate с ролью `closer`; если заданы `MPP2_PIPELINE_ID` и `MPP2_STATUS_TRANSFERRED`, сделка переставляется в воронку МПП-2 на клоузера.
- **Звонок в мобильное приложение ДОНУЛЯ** (`/api/app/provision`, `/call`, `/connect`): клиенту заводится SIP-аккаунт, бэкенд приложения получает запрос на push, после ответа в приложении Asterisk соединяет его через диалплан, а не через Stasis.

Входящих звонков в диалере нет: `StasisStart` без известной роли игнорируется.

Откуда данные и куда уходят

- **amoCRM** — **напрямую**, не через шлюз [amo-queue](#). Своя OAuth-интеграция: токены в таблице `settings`, обновляются по 401 и пишутся туда же. Все запросы идут через общий лимитер процесса (не чаще ~4 в секунду, пауза на 429/403). Читает сделки и контакты, пишет примечания о звонках, ответственного, этапы при переводе и в дедупликации.
- **PostgreSQL** — вся логика очередей, кампании, операторы, журнал звонков `call_log`, оценки, лента событий, настройки. База своя, отдельная.
- **Записи разговоров** лежат на диске сервера телефонии. Наружу они уходят только ссылкой `https://pbx.donula26.ru/api/recordings/<файл>` в примечании amoCRM: подпись HMAC (`RECORDINGS_HMAC_SECRET`) и срок жизни (`RECORDINGS_TTL_SEC`, 30 суток). Если wav уже сжат в opus, по той же ссылке отдаётся opus.
- **AI-разбор звонков МПП-2** — по вебхуку `/api/admin/webhook-analysis/<токен>` при переходе сделки в «Думает» или «Обещал оплатить»: склейка записей, распознавание (Yandex SpeechKit через Object Storage, запасной вариант AssemblyAI), анализ YandexGPT. Результат пишется в таблицу `call_analysis`. Включается настройкой `mpp_analysis_enabled` и флагом у менеджера.
- **TURN** — `turn.donula26.ru`: виджет и приложение берут временные кредиты в `/api/widget/turn-config` и `/api/app/turn-config`.

Связи с нашими сервисами:

Сервис	Как связан
mpp2-touches	pbx берёт у него rr-credits для выбора клоузера, сообщает о назначении менеджера и о смене статусов (<code>MPP2_TOUCHES_BASE</code>)
mpp2-tasks-widget	получает смену статусов менеджеров (<code>MPP2_TASKS_STATUS_URL</code>); сам ставит статус «Готов» / «Офф» через <code>POST /api/widget/agent-status</code> и читает график работы
dashboard, dashboard_mpp2, SpeechApi	забирают историю звонков из <code>/api/history</code> — API в формате команды <code>history</code> Мегафон OATC (<code>EXTERNAL_API_TOKEN</code>)
webhook_for_call_record	скачивает записи по ссылкам из примечаний amoCRM
бэкенд мобильного приложения, <code>app.donula26.ru</code>	заводит клиентам SIP-аккаунты в pbx и получает от pbx запрос на push о входящем (<code>APP_BACKEND_URL</code> , <code>APP_PROVISION_SECRET</code>)

Статусы всех менеджеров отдаются в `GET /api/manager-statuses` (`MANAGER_STATUS_TOKEN`): `ready` «Готов», `oncall` «Занят», `pause` «Пауза», `wrapup` «Постобработка», `learning` «Обучение», `offline` «Офф».

Где код

- Репозиторий `Buharestskie/pbx`, ветка `main`. Точка входа — `backend/src/app.js`, на проде процесс стартует через `backend/src/bootstrap.js`.
- Диалер — `backend/src/dialer/` (`agent-first.js` — основной сценарий, `transfer.js` — перевод, `originate.js`), события ARI разводятся в `app.js`.
- WebSocket — `backend/src/ws/gateway.js`, ручки — `backend/src/http/routes/`, amoCRM — `backend/src/integrations/amo/`, очередь и воркеры — `backend/src/queue/`, миграции — `backend/src/db/migrations/`.
- Переменные окружения читаются из `backend/.env`; полный список с описанием — в README. Часть настроек (токены amoCRM, `mpp_analysis_*`) живёт в таблице `settings`.

Локальный запуск

- Нужны Node.js 20+ и PostgreSQL. Asterisk не обязателен: без ARI backend пишет `Running in offline mode (no Asterisk)`, поднимает HTTP и WebSocket, но звонить не может.

2. Схема базы — только дампом схемы с прода (как снять — в README): миграции её с нуля не создают. Из `settings` удалить токены amoCRM, если локальный процесс не должен ходить в боевой amoCRM.
3. `cd backend && cp .env.example .env`, задать `PG_*`, любое значение `ASTERISK_AMI_PASSWORD` и локальные пути `PJSIP_AGENTS_PATH` / `PJSIP_APP_CLIENTS_PATH`.
4. `npm install`, `npm run migrate`, `npm start`, проверить `curl -s localhost:3000/health` — ответ `{"status":"ok","asterisk":"offline",...}`.
5. WebSocket — любым клиентом на `ws://localhost:3001`, первым сообщением `{"type":"auth","agent_id":...}`. Страницы `admin/` и `leadboard/` нужно открывать с того же адреса, что и API.

Выкатка

- **Только после 21:00 МСК.** Перезапуск бэкенда обрывает текущие разговоры, а днём простой телефонии видит вся компания. Читать логи можно в любое время.
- На сервере нет git: код копируется из `main` (rsync, без `--delete`). Перед этим — бэкап `scripts/pbx-backup.sh`. Дальше по обстоятельствам: `npm ci --omit=dev`, если менялись зависимости; `npm run migrate`, если появились миграции; перезапуск systemd-юнита `pbx-backend`, если менялся `backend/`. Правки `admin/`, `guide/`, `leadboard/` — статика, перезапуск не нужен.
- Проверка: `https://pbx.donula26.ru/health` отвечает `"asterisk":"connected"`, в журнале юнита — `Stasis app "pbx-dialer" started`. Виджеты сами переподнимают SIP примерно через 5 с после старта (сообщение `force_sip_reconnect`).
- Внешней пробы в мониторинге у pbx сейчас нет, поэтому окно обслуживания на статус-странице открыть не на что.

На что наступают

- **Виджета amoCRM в репозитории нет.** В amoCRM он загружен архивом, исходники не найдены. Меняя протокол WebSocket или `/api/widget/`, помните, что клиент поправить нечем.
- **Миграции не поднимают базу с нуля.** Части таблиц (`agent_groups`, `agent_campaigns`, `settings`, `trunks`, `transfers` и др.) нет ни в одной миграции, в нумерации дыры и дубли (два `028_*`, два `029_*`). На пустой базе `npm run migrate` падает.
- **Без `ASTERISK_AMI_PASSWORD` процесс падает на старте**, даже если Asterisk не нужен.
- **ARI подключается один раз, при старте.** Если Asterisk поднялся позже бэкенда, процесс жив, но не звонит; в `/health` — `"asterisk":"offline"`. Лечится перезапуском бэкенда после Asterisk.
- **Прод стартует `bootstrap.js`, а `npm start` — `app.js`.** `bootstrap.js` ограничивает параллельные HTTP-соединения к Asterisk шестью; локально этого ограничения нет. `src/config/app.js` — устаревшая копия `app.js`, её никто не импортирует.

- **Бэкенд должен жить на одном хосте с Asterisk.** Он сам пишет rjsip-конфиги, вызывает `asterisk -rx`, `ffmpeg/ffprobe`, а серверный путь к генератору отчёта постобработок зашит в код.
- `rsync --delete` **сотрёт то, чего нет в git:** установщики софтбона в `guide/app` и файлы, которые генерирует сервер. `admin/wrapup-report.html` и `admin/.wrapup-status.json` в git лежат старыми копиями — при выкатке их исключают, иначе затрётся свежий отчёт.
- **Кнопка «Перезапустить Asterisk» в админке** рвёт все звонки разом.
- **Поддомен amoCRM по умолчанию в коде** — `donulya2`, а админка и скрипты ссылаются на `donula.amocrm.ru`. `AMO_SUBDOMAIN` задавать явно.
- **Блокировка «Готов» работает в режиме fail-open:** если от amoCRM 5 минут нет вебхуков, пропускает всех. Аварийный откат — `READY_CHECK_ENABLED=0`.
- **Дедупликация лидов «Охотников» по умолчанию в dry-run** — решения только в логге. Реальные действия в amoCRM — с `DEDUP_DRY_RUN=0`.
- **Запись второго плеча перевода** по умолчанию пишется в другой каталог (`RECORDING_DIR`), чем тот, из которого раздаёт `/api/recordings`, — по ссылке её не получить.
- `.env.example` **не совпадает с кодом:** 8 переменных код не читает, около 20 читаемых там нет. Сверяйтесь с таблицей в README.
- **Ошибки в GlitchTip не уходят** — SDK в коде нет. Смотреть журнал юнита и `logs/error.log`.

Версия #1

Admin создал 2026-09-28 08:26:21 UTC

Admin обновил 2026-09-28 08:26:21 UTC