Сибирь.Диалог

Руководство по установке и эксплуатации

Версия 1.2.14. Редакция 4 от 30 октября 2026 года.

Правообладатель: ООО «Финансовый искусственный интеллект». ИНН 5402076457, ОГРН 1235400001870.

Архив программы: siberia-dialog-1.2.14.zip. Размер: 438 730 байт.

SHA-256:

```
ece57063f11f27e4e87fc50a68fc302d1f9fe7a28250d7894f411a1955a721ef
```

Руководство относится только к указанному архиву. Архив не изменен. В распакованной рабочей копии для установки создается пустой README.md, который требуется настройкой сборки пакета.

Документ описывает проверку целостности, установку, настройку, запуск, управление через API и командную строку, ответы программы, обработку ошибок, сохранение результатов и остановку.

Исторические примеры от 28 сентября 2026 года взяты из опубликованной документации правообладателя. Они сохраняют исходные даты и не являются результатами проверки 29 октября. Границы новой проверки приведены в заключительном разделе.

Установка зависимостей требует доступа к источнику пакетов. Модели, внешние службы, сценарии, лицензия и реквизиты подключения предоставляются отдельно. Режим проверки без телефонии использует SQLite и синтетический сценарий.

Публичная документация: https://siberia-ai.ru/dialog-docs.html. Поддержка: info@siberia-ai.ru.

Информация, необходимая для установки

Экземпляр поставляется архивом по лицензионному договору вместе с контрольной суммой SHA-256.

1. Проверка экземпляра

Экземпляр поставляется архивом siberia-dialog-1.2.14.zip размером 438 730 байта. До установки сверить контрольную сумму:

```
sha256sum siberia-dialog-1.2.14.zip
```

Ожидаемое значение: ece57063f11f27e4e87fc50a68fc302d1f9fe7a28250d7894f411a1955a721ef.

2. Требования к окружению

В исходной документации заявлена операционная система семейства Linux, включая Альт Сервер p11 и Astra Linux 1.8. Подтверждающие протоколы испытаний на этих ОС в данном комплекте не представлены. Новая выборочная проверка 29 октября выполнена на macOS и не подтверждает приемку Linux. Интерпретатор Python версии не ниже 3.12. СУБД PostgreSQL с расширением pgvector, хранилище Redis. Для голосового контура телефонная подсистема Asterisk и пакеты ffmpeg и ffprobe (в ОС Альт ffprobe устанавливается отдельным пакетом). Средства контейнеризации не обязательны.

3. Установка без телефонии для проверки работоспособности

Распаковать архив:

```
unzip siberia-dialog-1.2.14.zip -d dialog && cd dialog
```

Создать пустой файл README.md. Файл описания проекта pyproject.toml называет его в поле readme, а в архив включены только файлы исходного текста и файл описания проекта. Без README.md средство сборки прерывает установку ошибкой «OSError: Readme file does not exist: README.md».

```
touch README.md
```

Установить программу в отдельное окружение Python и активировать его:

```
python3.12 -m venv .venv
. .venv/bin/activate
python -m pip install -e . aiosqlite
```

Задать три обязательных значения и запустить сервер программного интерфейса. Значения ниже проверочные, для эксплуатации задаются свои. Для проверки без СУБД указана база SQLite, таблицы в ней программа создает сама:

```
export ARI_PASSWORD='check-ari-pass-12'
export SECRET_KEY='check-secret-key-0123456789abcdef'
export API_ADMIN_KEY='check-admin-key-0123456789'
export DATABASE_URL='sqlite+aiosqlite:///./dialog.db'
uvicorn siberia_dialog.api.app:app --port 8000
```

В журнале появляется строка «Application startup complete.». Проверка из другого окна:

```
curl -s http://127.0.0.1:8000/health
{"status":"ok"}
```

Проверка сценария и диалог в консоли. Сценарии диалогов в поставку не входят, их готовит эксплуатант и размещает в каталоге scenarios. Для проверки установки достаточно минимального сценария из двух состояний, его создают так:

```
mkdir -p scenarios
cat > scenarios/demo_info.yaml <<'EOF'
id: demo_info
vertical: generic
legal_basis: service
start: hello
slots: [full_name]
defaults:
  full_name: "Тестовый пользователь"
states:
  hello:
    say: "Здравствуйте, {full_name}. Это тестовое сообщение. Вам удобно говорить?"
    expect:
      intents: [yes, no]
    on:
      yes: bye
      no: bye
      _silence: bye
      _unknown: bye
  bye:
    say: "Спасибо, до свидания."
    end: informed
intents:
  yes: ["да", "удобно"]
  no: ["нет", "неудобно"]
EOF
python -m siberia_dialog.cli validate scenarios/demo_info.yaml
python -m siberia_dialog.cli chat scenarios/demo_info.yaml full_name="Иван Петрович"
```

Команда validate отвечает строкой «OK scenarios/demo_info.yaml: 2 состояний», команда chat ведет диалог в консоли, ответ вводится с клавиатуры, пример вывода приведен в разделе эксплуатации, п. 6.5.

4. Установка полного набора служб

Задать переменные окружения или записать их в файл .env в рабочем каталоге. Обязательны ARI_PASSWORD, SECRET_KEY и API_ADMIN_KEY. Подключение к СУБД и Redis задается строками DATABASE_URL и REDIS_URL, пароли входят в их состав. Прочие настройки перечислены в разделе эксплуатации, п. 6.2.

```
export DATABASE_URL='postgresql+asyncpg://dialog:<пароль>@localhost:5432/dialog'
export REDIS_URL='redis://:<пароль>@localhost:6379/0'
```

Применить миграции схемы базы данных. Файла alembic.ini в архиве нет, его создают в каталоге распаковки:

```
cat > alembic.ini <<'EOF'
[alembic]
script_location = %(here)s/src/siberia_dialog/db/migrations
[loggers]
keys = root
[handlers]
keys = console
[formatters]
keys = generic
[logger_root]
level = INFO
handlers = console
[handler_console]
class = StreamHandler
args = (sys.stderr,)
formatter = generic
[formatter_generic]
format = %(levelname)s %(message)s
EOF
alembic -c alembic.ini upgrade head
alembic -c alembic.ini current
```

Последняя команда отвечает «0012 (head)». Затем запустить службы, каждую отдельным процессом (подробно в разделе эксплуатации, п. 6.3):

```
uvicorn siberia_dialog.api.app:app --host 0.0.0.0 --port 8000
arq siberia_dialog.campaigns.worker.WorkerSettings
python -m siberia_dialog.voice.server
```

Голосовому обработчику нужна телефонная подсистема Asterisk с интерфейсом ARI. Те же команды можно выполнять средствами контейнеризации, это не обязательно.

5. Лицензия

Файл лицензии license.json передается лицензиату вместе с экземпляром и размещается в каталоге, указанном настройкой. Файл содержит наименование лицензиата, ИНН, срок действия и предельные значения: число одновременных голосовых линий и число чат-каналов. Подпись файла проверяется программой локально. Обращения к внешним службам активации не выполняются. Без файла лицензии, а также по истечении срока программа продолжает работать с демонстрационными ограничениями: две голосовые линии и один чат-канал. Статус лицензии виден в сведениях о программе.

Информация, необходимая для эксплуатации

Руководство администратора: загрузка, запуск, выполнение и завершение программы, команды управления и ответы программы на них.

Относится к экземпляру siberia-dialog-1.2.14.zip. Примеры ответов получены запуском этого экземпляра 28 сентября 2026 года с базой SQLite, без телефонной подсистемы, Redis и PostgreSQL. Где запуск не выполнялся, это сказано в тексте. Ответы и сообщения программы приведены дословно, с ее собственной орфографией и знаками.

6. Эксплуатация

6.1. Состав программы

Программа устанавливается как пакет Python siberia_dialog и работает несколькими процессами. Все процессы читают одни и те же настройки из переменных окружения и из файла .env в текущем каталоге процесса.

Сервер программного интерфейса

Команда запуска: `uvicorn siberia_dialog.api.app:app`

Назначение: Управление заказчиками, кампаниями, согласиями и отказами, базой знаний, документами, отчетами. Прием вебхуков чат-каналов

Обработчик кампаний

Команда запуска: `arq siberia_dialog.campaigns.worker.WorkerSettings`

Назначение: Набор номеров каждые 10 секунд, шаги лестницы касаний, импорт баз, удаление данных по срокам хранения, разбор записей речевой аналитикой

Голосовой обработчик

Команда запуска: `python -m siberia_dialog.voice.server`

Назначение: Прием событий Asterisk ARI и аудиопотока AudioSocket, голосовой диалог

Адаптер Telegram

Команда запуска: `python -m siberia_dialog.channels.telegram`

Назначение: Бот Telegram опросом сервера Telegram, если вебхук не используется

Командная строка

Команда запуска: `python -m siberia_dialog.cli`

Назначение: Проверка и отладка сценариев, прогрев кэша синтеза речи, перешифрование секретов, паспорт сценария, самопроверка

Проверка лицензии

Команда запуска: `python -m siberia_dialog.cli_license`

Назначение: Проверка подписи файла лицензии

Проверка живости

Команда запуска: `python -m siberia_dialog.healthcheck worker|voice`

Назначение: Код 0, если обработчик кампаний или голосовой обработчик жив

Внешние службы в состав экземпляра не входят и устанавливаются эксплуатантом: СУБД PostgreSQL, хранилище Redis, телефонная подсистема Asterisk с интерфейсом ARI, пакеты ffmpeg и ffprobe для речевой аналитики, программа tesseract для распознавания документов. Сценарии диалогов, правила речевой аналитики, шаблоны документов, модели распознавания и файл лицензии размещает эксплуатант в каталогах, заданных настройками.

6.2. Загрузка

Программа загружается по разделу установки: проверка контрольной суммы, распаковка, создание файла README.md, установка пакета, настройка окружения, миграции (пп. 1, 3, 4). Ответ программы установки, если шаг с README.md пропущен (проверено 28.09.2026):

```
OSError: Readme file does not exist: README.md
```

Дополнения, указываемые в квадратных скобках при установке: voice для голосового обработчика, chat для адаптеров Telegram и VK, rag для векторного поиска по базе знаний, dev для средств разработки.

Обязательные переменные окружения. Без них не запускается ни один процесс программы:

`ARI_PASSWORD`

Требование: Не короче 12 знаков

Назначение: Пароль пользователя ARI в Asterisk

`SECRET_KEY`

Требование: Не короче 32 знаков

Назначение: Шифрование токенов заказчиков, подпись сеансов веб-виджета

`API_ADMIN_KEY`

Требование: Не короче 24 знаков

Назначение: Ключ администратора для заголовка X-API-Key

Ответ программы, если они не заданы (код возврата 1):

```
pydantic_core._pydantic_core.ValidationError: 3 validation errors for Settings
ari_password
  Field required [type=missing, input_value={}, input_type=dict]
secret_key
  Field required [type=missing, input_value={}, input_type=dict]
api_admin_key
  Field required [type=missing, input_value={}, input_type=dict]
```

Основные необязательные переменные и значения по умолчанию:

`DATABASE_URL`

По умолчанию: `postgresql+asyncpg://dialog:dialog@localhost:5432/dialog`

Назначение: Строка подключения к СУБД, пароль СУБД задается в ее составе

`REDIS_URL`

По умолчанию: `redis://localhost:6379/0`

Назначение: Очередь обработчика кампаний, пароль Redis задается в составе строки

`ARI_URL`, `ARI_USER`, `ARI_APP`

По умолчанию: `http://localhost:8088/ari`, dialog, siberia_dialog

Назначение: Подключение к Asterisk

`AUDIOSOCKET_HOST`, `AUDIOSOCKET_PORT`

По умолчанию: 0.0.0.0, 9092

Назначение: Прием AudioSocket голосовым обработчиком

`STT_PROVIDER`, `TTS_PROVIDER`

По умолчанию: yandex, yandex

Назначение: Распознавание (yandex, salute, vosk) и синтез (yandex, salute) речи

`YANDEX_API_KEY`, `SALUTE_CLIENT_ID`, `SALUTE_CLIENT_SECRET`

По умолчанию: Пусто

Назначение: Реквизиты речевых служб

`GATEWAY_URL`, `GATEWAY_API_KEY`

По умолчанию: Пусто

Назначение: Языковая модель. Пусто означает работу на правилах и словарях

`LICENSE_PATH`

По умолчанию: license.json

Назначение: Файл лицензии

`SCENARIOS_DIR`

По умолчанию: ./scenarios

Назначение: Каталог сценариев <scenario_id>.yaml

`RECORDINGS_DIR`, `DATA_DIR`, `TTS_CACHE_DIR`

По умолчанию: ./recordings, ./data, ./tts_cache

Назначение: Записи разговоров, данные, кэш синтеза

`ANALYTICS_RULES_DIR`, `DOCUMENT_TEMPLATES_DIR`

По умолчанию: ./analytics_rules, ./document_templates

Назначение: Правила речевой аналитики, шаблоны документов

`ENABLE_DOCS`

По умолчанию: true

Назначение: Страница описания программного интерфейса /docs

`MAX_UPLOAD_MB`

По умолчанию: 20

Назначение: Предельный размер загружаемого файла

RATE_LIMIT_ADMIN_PER_MIN, RATE_LIMIT_OCR_PER_MIN, RATE_LIMIT_WEBHOOK_PER_MIN, RATE_LIMIT_WEBHOOK_IP_PER_MIN

По умолчанию: 600, 30, 60, 600

Назначение: Лимиты запросов в минуту

RATE_LIMIT_BACKEND, SESSION_BACKEND

По умолчанию: memory, memory

Назначение: Хранение окон лимитов и сеансов чатов: memory или redis

TRUSTED_PROXIES

По умолчанию: Пусто

Назначение: Прокси, чьему заголовку X-Forwarded-For программа доверяет

SINGLE_TENANT

По умолчанию: true

Назначение: Один заказчик на инсталляцию

TRUNK_CHANNELS

По умолчанию: 0

Назначение: Емкость SIP-транка, 0 означает, что общий предел не применяется

CALL_WATCHDOG_SEC

По умолчанию: 90

Назначение: Через сколько секунд после начала вызова проверять, жив ли канал

OPTOUT_REGISTRY_REQUIRED, OPTOUT_REGISTRY_MAX_AGE_DAYS

По умолчанию: true, 30

Назначение: Обязательность и срок свежести сверки с реестром отказов оператора

LIMIT_PER_DAY, LIMIT_PER_WEEK, LIMIT_PER_MONTH

По умолчанию: 1, 2, 8

Назначение: Частотные ограничения попыток

WINDOW_WEEKDAY, WINDOW_WEEKEND

По умолчанию: 08:00-22:00, 09:00-20:00

Назначение: Допустимые окна вызовов

PUBLIC_BASE_URL

По умолчанию: Пусто

Назначение: Внешний адрес программы, нужен для подписки вебхука MAX

EXECUTOR_NAME, EXECUTOR_ADDRESS, SYSTEM_LOCATION

По умолчанию: ООО «Финансовый искусственный интеллект», пусто, пусто

Назначение: Реквизиты в акте уничтожения персональных данных

ANALYTICS_RETENTION_DAYS

По умолчанию: 30

Назначение: Срок хранения записей пакета аналитики после передачи отчета

Схема базы данных. При строке подключения к SQLite сервер программного интерфейса создает таблицы сам при запуске. Для PostgreSQL схема создается миграциями Alembic из каталога src/siberia_dialog/db/migrations, последняя ревизия 0012. Адрес базы миграции берут из DATABASE_URL. Файла alembic.ini в архиве нет, он создается при установке (раздел установки, п. 4). Миграции проверены на SQLite, на PostgreSQL не проверялись:

```
alembic -c alembic.ini upgrade head
alembic -c alembic.ini current
```

Ниже приведен исторический вывод указанной команды. Вывод программы не вводится в оболочку.

```
0012 (head)
```

6.3. Запуск

Порядок: СУБД и Redis, миграции, сервер программного интерфейса, обработчик кампаний, голосовой обработчик. Адрес и порт сервера выбирает эксплуатант, 8000 в примерах условный. Команды выполняются в окружении, активированном после установки командой . .venv/bin/activate.

Сервер программного интерфейса.

```
uvicorn siberia_dialog.api.app:app --host 0.0.0.0 --port 8000
```

При запуске программа проверяет базу часовых поясов (без нее запуск прерывается), языковые пакеты tesseract (при их отсутствии пишет предупреждение и продолжает работу) и для SQLite создает таблицы. Исторический пример журнала от 28 сентября 2026 года относится к конфигурации Windows с портом 18770. В том прогоне tesseract не был установлен. Порт 8000 в командах выше является условным примером, для своей установки используйте выбранный порт:

```
INFO:     Started server process [17348]
INFO:     Waiting for application startup.
распознавание документов недоступно: tesseract не запускается: [WinError 2] The system cannot find the
file specified
INFO:     Application startup complete.
INFO:     Uvicorn running on http://127.0.0.1:18770 (Press CTRL+C to quit)
```

Проверка работоспособности:

```
curl -s http://127.0.0.1:8000/health
curl -s http://127.0.0.1:8000/about
```

Ниже приведен исторический вывод указанной команды. Вывод программы не вводится в оболочку.

```
{"status":"ok"}
{"product":"Siberia.Dialog","version":"1.2.14","rightsholder":{"name":"ООО «Финансовый искусственный
интеллект» (Siberia AI)","inn":"5402076457","kpp":"547301001","ogrn":"1235400001870","address":"630090,
г. Новосибирск, ул. Демакова, д. 23/5","site":"https://siberia-ai.ru","support":"support@siberia-ai.ru,
+7 (383) 288-82-82"},"license":{"licensee":"демонстрационный
режим","inn":"","valid_until":"","limits":{"voice_lines":2,"chat_channels":1,"documents":5},"status":"demo"
},"third_party":"см. LICENSE-THIRD-PARTY.md в составе дистрибутива","registry_class":"05.09 Средства
управления диалоговыми роботами","usage":{"voice_lines_in_use":0,"voice_lines_limit":2,"chat_channels_in_u
se":[],"chat_channels_limit":1}}
```

Поле license.status принимает значения valid, expired, invalid и demo. Без файла лицензии, с недействительной подписью и по истечении срока действуют демонстрационные ограничения: две голосовые линии, один чат-канал, пять страниц документа на запрос.

Обработчик кампаний.

```
arq siberia_dialog.campaigns.worker.WorkerSettings
```

Расписание: набор номеров каждые 10 секунд, шаги лестницы касаний каждую минуту, импорт баз в 03:00, удаление данных по срокам хранения в 04:00, повторная постановка застрявших пакетов аналитики каждые 5 минут. Без ffmpeg обработчик пишет предупреждение «речевая аналитика недоступна» и продолжает обзвон. Запуск с Redis в прогоне не выполнялся. Без Redis обработчик делает пять попыток подключения и завершается с кодом 1:

```
01:58:26: redis connection error 127.0.0.1:16379 TimeoutError Timeout connecting to server, 5 retries
remaining...
...
01:58:34: redis connection error 127.0.0.1:16379 TimeoutError Timeout connecting to server, 1 retries
remaining...
```

Проверка живости (обработчик обновляет отметку в Redis раз в 30 секунд). Код 0 означает, что обработчик жив, код 1 означает, что нет, причина печатается в поток ошибок:

```
REDIS_URL=redis://127.0.0.1:16379/0 python -m siberia_dialog.healthcheck worker
```

Ниже приведен исторический вывод указанной команды. Вывод программы не вводится в оболочку.

```
проверка живости worker: ConnectionError('Error 10061 connecting to 127.0.0.1:16379. No connection could
be made because the target machine actively refused it.')
```

Голосовой обработчик.

```
python -m siberia_dialog.voice.server
```

Признак запуска в журнале:

```
2026-09-28 01:58:01,494 voice INFO AudioSocket on 127.0.0.1:19092
```

Эта строка и проверка живости подтверждают только прием AudioSocket. Подключение к Asterisk программа в журнал не пишет: при недоступном ARI она повторяет подключение каждые 2 секунды без сообщений. В прогоне Asterisk не было, а проверка живости ответила кодом 0:

```
AUDIOSOCKET_PORT=19092 python -m siberia_dialog.healthcheck voice
echo $?
```

Ниже приведен исторический вывод указанной команды. Вывод программы не вводится в оболочку.

```
0
```

Если порт AudioSocket занять не удалось, процесс не завершается, а пишет ошибку «AudioSocket ... не поднялся» и повторяет попытку через 30 секунд. Работа с Asterisk и голосовой диалог в прогоне не проверялись.

Адаптер Telegram опросом. Запускается командой python -m siberia_dialog.channels.telegram при заданном TELEGRAM_BOT_TOKEN. В прогоне не запускался.

6.4. Выполнение: программный интерфейс

Ответы в таблице ниже являются историческими примерами прогона 28 сентября 2026 года. В примерах веб-виджета параметр full_name не был заполнен, поэтому после приветствия показано пустое имя. Новый минимальный сценарий из раздела установки содержит значение full_name по умолчанию. Исторические ответы не выдаются за результат этого нового сценария.

6.4.1. Общие правила. Запросы управления сопровождаются заголовком X-API-Key. Ключ не проверяется у адресов /health, /about, /docs, /openapi.json, /metrics и у вебхуков /webhooks/..., которые проверяют секреты каналов. Ключ администратора задается в API_ADMIN_KEY, ключ заказчика выдается запросом POST /customers/{customer_id}/api-key и хранится в базе только в виде хэша. Заказчик видит и меняет только свои данные. Описание программного интерфейса в формате OpenAPI отдается по адресу /openapi.json, интерактивная страница по адресу /docs, метрики по адресу /metrics.

Ответ об ошибке содержит поле detail. Ошибки проверки входных данных приходят с кодом 422 списком:

```
{"detail":[{"type":"less_than_equal","loc":["body","concurrency"],"msg":"Input should be less than or
equal to 200","input":500,"ctx":{"le":200}}]}
```

401

detail: `X-API-Key required`

Причина: Нет заголовка ключа

401

detail: `invalid API key`

Причина: Ключ не совпадает ни с ключом администратора, ни с ключом заказчика

403

detail: `admin role required`

Причина: Операция только для администратора

403

detail: `access to another customer denied`

Причина: Ключ заказчика обращается к чужому заказчику

403

detail: `access to another customer's campaign denied`

Причина: Ключ заказчика обращается к чужой кампании

404

detail: `Not Found`

Причина: Объекта нет. Для несуществующей кампании 404 получает и администратор

413

detail: `file too large (> 20 MB)`

Причина: Файл больше MAX_UPLOAD_MB

400

detail: `empty file`, `unsupported file type`, `damaged ZIP container`

Причина: Пустой файл, формат не принимается этой операцией, поврежденный архив

429

detail: `rate limit exceeded`

Причина: Превышен лимит запросов, заголовок Retry-After: 60. Окно считается на ключ (для документов отдельное), для вебхуков на заказчика и адрес и отдельно на адрес

502

detail: `upstream service error`

Причина: Внешняя служба ответила ошибкой

Тип загружаемого файла определяется по содержимому, а не по расширению. Каждый запрос POST, PUT, PATCH, DELETE вне /webhooks/ записывается в журнал аудита с ролью, адресом и кодом ответа.

6.4.2. Порядок работы с кампанией.

1. Создать заказчика (администратор)

Запрос: `POST /customers`

Ответ программы в прогоне: 200: `{"id":"2d017d916c8a403c8d5f110281501855"}`

2. Выдать ключ заказчика

Запрос: `POST /customers/{customer_id}/api-key`

Ответ программы в прогоне: 200: `{"api_key":"sd_..."}`

3. Для голосовой кампании заполнить маркировку вызовов

Запрос: `PATCH /customers/{customer_id}/settings с call_labeling`

Ответ программы в прогоне: 200:
{"settings":{"chat_scenario":"demo_info","call_labeling":{"org_name":"УК Пример","category":"info"},"channels":{"max":{"secret":"***"}}}}

4. Загрузить выгрузку реестра отказов оператора (администратор)

Запрос: `POST /compliance/optout-registry/import`

Ответ программы в прогоне: 200: {"rows":2,"invalid":1,"added":2,"source":"operator","provider":"Novofon","exported_at":"2026-09-20T00:00:00+00:00","digest":"d35014feef7287081d59bf17e3775c9e231119aeca4b9a178db6c5ba06dd33c1"}

5. Создать кампанию, файл сценария лежит в SCENARIOS_DIR

Запрос: `POST /campaigns`

Ответ программы в прогоне: 200: {"id":"77fd896bbd964b9cbc10f6323c39b4f8","legal_basis":"service"}

6. Загрузить контакты CSV или XLSX

Запрос: `POST /campaigns/{campaign_id}/contacts`

Ответ программы в прогоне: 200: {"loaded":1,"errors":["строка 3: некорректный телефон «12345»"],"errors_total":1}

7. Запустить

Запрос: `POST /campaigns/{campaign_id}/start`

Ответ программы в прогоне: 200: {"id":"77fd896bbd964b9cbc10f6323c39b4f8","status":"running"}

8. Следить за ходом

Запрос: `GET /campaigns/{campaign_id}`

Ответ программы в прогоне: 200: {"id":"77fd896bbd964b9cbc10f6323c39b4f8","name":"Чат","status":"running","scenario_id":"demo_info","channel":"telegram","legal_basis":"service","contacts":{"pending":1}}

9. Приостановить, возобновить повторным start

Запрос: `POST /campaigns/{campaign_id}/pause`

Ответ программы в прогоне: 200: {"id":"77fd896bbd964b9cbc10f6323c39b4f8","status":"paused"}

10. Получить отчеты и акты

Запрос: `GET /reports/campaigns/{campaign_id}, /xlsx, /csv, /losses, /ab, /executive, GET /reports/attempts/{attempt_id}/proof`

Ответ программы в прогоне: 200: {"campaign":"Чат","status":"running","contacts":1,"attempts":0,"reached":0,"reach_rate":0.0,"promised":0,"promised_sum_rub":0,"claims_paid":0,"dnc":0,"outcomes":{},"contacts_by_outcome":{},"talk_minutes":0.0,"cost_total_rub":null,"cost_measured_attempts":0,"cost_per_reached_rub":null,"by_status":{"p ...

Статусы кампании: draft после создания, running, paused, done. Голосовая кампания переходит в done сама, когда у нее не остается контактов в статусах pending, scheduled, calling. Статусы контакта: pending, calling, scheduled (назначен повтор), done, failed, dnc, skipped.

6.4.3. Отказы запуска кампании. Все с кодом 409, причина в поле detail:

`реклама автодозвоном и автоматической рассылкой запрещена ...`

Причина: Основание advertising у кампании с автоматическим набором

Что делать: Сменить основание или канал

`реестр отказов оператора не загружен: ...,выгрузка реестра отказов от ... устарела ...`

Причина: Нет свежей выгрузки реестра отказов (кроме входящей линии voice_in)

Что делать: Загрузить выгрузку, шаг 4

`{"detail":"речь не настроена, кампания не запускается: TTS yandex: не задан YANDEX_API_KEY; STT yandex: не задан YANDEX_API_KEY"}`

Причина: Для voice и voice_in не заданы реквизиты синтеза или распознавания речи. Ответ получен в прогоне

Что делать: Задать YANDEX_API_KEY или реквизиты SaluteSpeech

`сценарий ... не найден` **и другие ошибки настройки сравнения или сегментов**

Причина: Нет файла сценария варианта или он другой отрасли

Что делать: Исправить настройку кампании или положить файл сценария

`маркировка вызовов не заявлена: заполните settings.call_labeling (org_name, category) ...`

Причина: Для voice не заполнена маркировка вызовов

Что делать: Шаг 3

`неизвестная категория маркировки ...,кампания заявлена рекламной ...`

Причина: Неверная категория или ее расхождение с основанием

Что делать: Исправить call_labeling.category

6.4.4. Перечень операций. Всего 73 операции. Доступ: А администратор, З заказчик в пределах своих данных. Коды 401 и 429 возможны у всех операций с ключом и в таблице не повторяются. Все 73 операции вызваны в прогоне 28.09.2026, у 67 получен успешный ответ, у 6 только ответ об ошибке с указанной причиной. Коды ошибок в таблицах взяты из исходного текста программы, примеры ответов из прогона.

Служебные адреса

GET `/health`

Назначение: Проверка работоспособности. Ключ не нужен

Формат запроса: Нет

Пример ответа: 200: `{"status":"ok"}`

Коды ошибок: Нет

GET `/health/providers`

Назначение: Счетчики обращений к внешним службам с момента запуска процесса. Доступ: А

Формат запроса: Нет

Пример ответа: 200: `{"providers":{}}`

Коды ошибок: 401, 403

GET `/about`

Назначение: Сведения о программе, правообладателе, лицензии и использовании лимитов. Ключ не нужен

Формат запроса: Нет

Пример ответа: 200: `{"product":"Siberia.Dialog","version":"1.2.14","rightsholder":{"name":"ООО «Финансовый искусственный интеллект» (Siberia AI)","inn":"5402076457","kpp":"547301001","ogrn":"1235400001870","address":"630090, г. Новосибирск, ул. Демак ...`

Коды ошибок: Нет

Заказчики

POST `/customers`

Назначение: Создать заказчика. Доступ: A

Формат запроса: JSON: `name` 1-255 знаков, обязательно, `inn` (проверяется контрольная сумма), `phone` до 32 знаков, `vertical` zhkh, energy, clinic или generic, `tz` часовой пояс IANA (по умолчанию Europe/Moscow), `in_gov_list`, `settings`

Пример ответа: 200: `{"id":"2d017d916c8a403c8d5f110281501855"}`

Коды ошибок: 422 (неизвестный часовой пояс, неверный ИНН)

GET `/customers`

Назначение: Перечень заказчиков. Доступ: A

Формат запроса: Нет

Пример ответа: 200: `[{"id":"2d017d916c8a403c8d5f110281501855","name":"УК Пример","vertical":"zhkh","tz":"Asia/Novosibirsk"},{"id":"53a12e3a29434f57b5e0e8b175995205","name":"УК Вторая","vertical":"zhkh","tz":"Europe/Moscow"}]`

Коды ошибок: 403 для ключа заказчика

GET `/customers/{customer_id}`

Назначение: Карточка заказчика, секреты показаны маской. Доступ: A, З

Формат запроса: Нет

Пример ответа: 200: `{"id":"2d017d916c8a403c8d5f110281501855","name":"УК Пример","inn":"5402076457","phone":null,"vertical":"zhkh","tz":"Asia/Novosibirsk","in_gov_list":false,"settings":{},"has_api_key":false}`

Коды ошибок: 403, 404

PATCH `/customers/{customer_id}/settings`

Назначение: Дополнить настройки заказчика. Ключи верхнего уровня объединяются с прежними, вложенный объект (например `channels`) заменяется целиком. Доступ: A

Формат запроса: JSON: `channels.telegram|max|vk|sms|voice`, `call_labeling` (`org_name`, `category`: info, service, survey, ads, collection), `chat_scenario`

Пример ответа: 200: `{"settings":{"chat_scenario":"demo_info","call_labeling":{"org_name":"УК Пример","category":"info"},"channels":{"max":{"secret":"***"}}}}`

Коды ошибок: 404, 409 (превышен лимит чат-каналов лицензии)

POST `/customers/{customer_id}/api-key`

Назначение: Выпустить новый ключ заказчика. Показывается один раз, прежний перестает действовать. Доступ: A

Формат запроса: Нет

Пример ответа: 200: `{"api_key":"sd_..."}`

Коды ошибок: 404

POST `/customers/{customer_id}/base`

Назначение: Загрузить базу абонентов и приборов учета. Доступ: A, З

Формат запроса: Форма, поле `file`: XLSX с колонками Лицевой счет, ФИО, Тип, Улица, Дом, Корпус, Квартира, Телефон, Задолженность, Услуга, Заводской номер, Дата поверки, Предыдущее показание

Пример ответа: 200: `{"accounts":1,"meters":1}`

Коды ошибок: 400 (неверная база или тип файла), 404, 413, 500 (нет прав на каталог данных, причина в тексте ответа)

GET `/customers/{customer_id}/media/{ref}`

Назначение: Фото, присланное абонентом. Доступ: A, З

Формат запроса: Нет

Пример ответа: Успешный ответ в прогоне не получен: фото не присылались. Ответ: 404: `{"detail":"изображение не найдено"}`

Коды ошибок: 404

GET `/customers/{customer_id}/readings.xlsx`

Назначение: Реестр принятых показаний. Доступ: A, 3

Формат запроса: Нет

Пример ответа: 200: книга XLSX 5 031 байт

Коды ошибок: 404 (база не загружена)

POST `/customers/{customer_id}/close`

Назначение: Закрыть пилот: выгрузка данных оператору, удаление и обезличивание данных жителей, акт. Повторный вызов возвращает тот же акт. Доступ: A

Формат запроса: JSON: `signed_by` 3-255 знаков, `signed_title` 2-255 знаков, `transfer_act` до 255 знаков

Пример ответа: 200: `{"act":{"customer_id":"53a12e3a29434f57b5e0e8b175995205","closed_at":"2026-09-28T08:57:19+00:00","by":"admin:admin","export":"53a12e3a29434f57b5e0e8b175995205-20260928085719.zip","export_entries":["consents.csv","dnc.csv","subject ...`

Коды ошибок: 404, 422

POST `/customers/{customer_id}/close/transfer-confirmed`

Назначение: Отметить, что оператор получил выгрузку. Копия выгрузки удаляется, отметка вносится в акт. Доступ: A

Формат запроса: Нет

Пример ответа: 200: `{"act":{"customer_id":"53a12e3a29434f57b5e0e8b175995205","closed_at":"2026-09-28T08:57:19+00:00","by":"admin:admin","export":"53a12e3a29434f57b5e0e8b175995205-20260928085719.zip","export_entries":["consents.csv","dnc.csv","subject ...`

Коды ошибок: 404, 409

Отказы и согласия заказчика (доступ: A, 3)

GET `/customers/{customer_id}/dnc`

Назначение: Перечень отказов от взаимодействия

Формат запроса: Нет

Пример ответа: 200: `[{"phone":"79131234567","source":"operator","created_at":"2026-09-28T08:57:03"}]`

Коды ошибок: 403

POST `/customers/{customer_id}/dnc`

Назначение: Добавить отказ

Формат запроса: JSON: `phone` до 32 знаков, `source` call, operator, import, gosuslugi, contract, lk, sms или paper, `evidence` до 1000 знаков

Пример ответа: 200: `{"status":"added","keys":["79131234567"]}`

Коды ошибок: 400 (`bad phone`), 404

DELETE `/customers/{customer_id}/dnc/{phone}`

Назначение: Снять отказ по всем связанным идентификаторам

Формат запроса: Нет

Пример ответа: 200: `{"removed":1}`

Коды ошибок: 400

POST `/customers/{customer_id}/dnc/import`

Назначение: Массовый импорт отказов

Формат запроса: Форма, поле `file`: CSV или XLSX с колонкой телефона

Пример ответа: 200: `{"imported":2}`

Коды ошибок: 400, 404, 413, 500 (CSV из одной колонки, см. п. 6.7)

POST `/customers/{customer_id}/identity-links`

Назначение: Связать телефон с идентификатором абонента в канале. Отказ распространяется на обе стороны

Формат запроса: JSON: `phone`, `channel_identity` вида канал:идентификатор

Пример ответа: 200: `{"created":true,"identity_links":1}`

Коды ошибок: 400, 404, 422

POST `/customers/{customer_id}/consents`

Назначение: Записать согласие

Формат запроса: JSON: `phone`, `purpose` pdn, service или advertising (по умолчанию service), `source`, `evidence`

Пример ответа: 200: `{"status":"added","purpose":"service"}`

Коды ошибок: 400, 404

GET `/customers/{customer_id}/consents`

Назначение: Перечень согласий с датами записи и отзыва

Формат запроса: Нет

Пример ответа: 200: `[{"phone":"79139876543","purpose":"service","source":"contract","created_at":"2026-09-28T08:57:04","revoked_at":null}]`

Коды ошибок: 403

GET `/customers/{customer_id}/consents/{phone}`

Назначение: Действующие цели согласия абонента

Формат запроса: Нет

Пример ответа: 200: `{"has_consent":true,"purposes":["service"],"advertising":false}`

Коды ошибок: 400

DELETE `/customers/{customer_id}/consents/{phone}`

Назначение: Отозвать согласие. Без параметра отзываются все цели, запись сохраняется

Формат запроса: Параметр `purpose` (необязателен)

Пример ответа: 200: `{"revoked":["service"]}`

Коды ошибок: 400

Реестр отказов оператора (доступ: A)

GET `/compliance/optout-registry`

Назначение: Состояние сверки с реестром отказов оператора

Формат запроса: Нет

Пример ответа: 200: `{"required":true,"max_age_days":30,"fresh":true,"problem":null,"last_import":{"source":"operator","provider":"Novofon","exported_at":"2026-09-20T00:00:00","imported_at":"2026-09-28T08:57:04","age_days":8,"rows":2,"invalid":1,"adde ...`

Коды ошибок: 403

GET `/compliance/operator/lines`

Назначение: SIP-линии и виртуальные номера у оператора связи (Data API Новофон)

Формат запроса: Нет

Пример ответа: Успешный ответ в прогоне не получен: ключ оператора не задавался. Ответ: 409: `{"detail":"не задан NOVOFON_API_KEY: ключ Data API оператора связи"}`

Коды ошибок: 409 (нет `NOVOFON_API_KEY`), 502 (оператор не ответил)

POST `/compliance/optout-registry/import`

Назначение: Загрузить выгрузку реестра отказов. Повторная загрузка того же файла дает added: 0

Формат запроса: Форма: `file` CSV, XLSX или TXT, `source` operator, gosuslugi или rkn, `provider` до 128 знаков, `exported_at` дата ГГГГ-ММ-ДД

Пример ответа: 200: `{"rows":2,"invalid":1,"added":2,"source":"operator","provider":"Novofon","exported_at":"2026-09-20T00:00:00+00:00","digest":"d35014feef7287081d59bf17e3775c9e231119aeca4b9a178db6c5ba06dd33c1"}`

Коды ошибок: 400, 413, 422

База знаний (доступ: A, 3)

POST `/customers/{customer_id}/knowledge/collections`

Назначение: Создать коллекцию

Формат запроса: JSON: `name` 1-255 знаков, `description` до 2000 знаков

Пример ответа: 201: `{"id":"a164b9a044fb40559278badd6f092e72","name":"Регламенты","version":1}`

Коды ошибок: 404, 422

GET `/customers/{customer_id}/knowledge/collections`

Назначение: Перечень коллекций

Формат запроса: Нет

Пример ответа: 200: `[{"id":"a164b9a044fb40559278badd6f092e72","name":"Регламенты","description":null,"version":1,"documents":0,"chunks":0,"created_at":"2026-09-28T08:57:04"}]`

Коды ошибок: 403

DELETE `/customers/{customer_id}/knowledge/collections/{collection_id}`

Назначение: Удалить коллекцию с документами

Формат запроса: Нет

Пример ответа: 204, тело пустое

Коды ошибок: 404

GET `/customers/{customer_id}/knowledge/{collection_id}/documents`

Назначение: Перечень документов коллекции

Формат запроса: Нет

Пример ответа: 200: `[{"id":"b6eaee7f4a2a4edf9b6d4e94b78b3d0a","title":"kb.txt","filename":"kb.txt","kind":"txt","pages":null,"chunks":1,"size_bytes":80,"sha256":"ee887318c33f140be45504ecf9c3562bd4fb5de5a7058d1321203941e51e0007","status":"ready","uplo ...`

Коды ошибок: 404

POST `/customers/{customer_id}/knowledge/{collection_id}/documents`

Назначение: Загрузить и проиндексировать документ

Формат запроса: Форма, поле `file`: PDF, DOCX, HTML, MD или TXT, параметр `title`

Пример ответа: 201: `{"id":"b6eaee7f4a2a4edf9b6d4e94b78b3d0a","title":"kb.txt","kind":"txt","pages":null,"chunks":1,"collection_version":2}`

Коды ошибок: 400 (документ не разобран), 404, 413

DELETE `/customers/{customer_id}/knowledge/{collection_id}/documents/{document_id}`

Назначение: Удалить документ

Формат запроса: Нет

Пример ответа: 204, тело пустое

Коды ошибок: 404

GET `/customers/{customer_id}/knowledge/{collection_id}/documents/{document_id}/source`

Назначение: Исходный файл документа

Формат запроса: Нет

Пример ответа: 200: `Оплата квитанции до 10 числа каждого месяца.`

Коды ошибок: 404

POST `/customers/{customer_id}/knowledge/{collection_id}/query`

Назначение: Ответ на вопрос по коллекции с источниками. Если ответа нет, так и сообщается

Формат запроса: JSON: `question` 1-2000 знаков, `top_k` 1-20

Пример ответа: 200: `{"answer":"Оплата квитанции до 10 числа каждого месяца.\n- kb.txt, фрагмент 1","found":true,"generated":false,"sources":[{"document_id":"b6eaee7f4a2a4edf9b6d4e94b78b3d0a","document_title":"kb.txt","citation":"фрагмент 1","page":nu ...`

Коды ошибок: 404 (`коллекция не найдена`)

Кампании

POST `/campaigns`

Назначение: Создать кампанию. Доступ: любой ключ, заказчик только для себя

Формат запроса: JSON: `customer_id`, `name`, `scenario_id` (файл `<scenario_id>.yaml` в каталоге сценариев), `channel` voice, telegram, max, vk или voice_in, `legal_basis`, `concurrency` 1-200, `max_attempts` 1-10, `defaults`

Пример ответа: 200: `{"id":"77fd896bbd964b9cbc10f6323c39b4f8","legal_basis":"service"}`

Коды ошибок: 400 (сценарий не найден), 403, 404, 422

POST `/campaigns/{campaign_id}/contacts`

Назначение: Загрузить контакты. Доступ: A, З своей кампании

Формат запроса: Форма, поле `file`: CSV или XLSX, колонка телефона phone, телефон, тел или номер

Пример ответа: 200: `{"loaded":1,"errors":["строка 3: некорректный телефон «12345»"],"errors_total":1}`

Коды ошибок: 400 (тип файла, больше 500 000 строк), 404, 413, 500 (CSV из одной колонки)

POST `/campaigns/{campaign_id}/start`

Назначение: Запустить или возобновить кампанию. Доступ: A, З своей кампании

Формат запроса: Нет

Пример ответа: 200: `{"id":"77fd896bbd964b9cbc10f6323c39b4f8","status":"running"}`

Коды ошибок: 404, 409 (причина в тексте, см. п. 6.4.3)

POST `/campaigns/{campaign_id}/pause`

Назначение: Приостановить кампанию. Доступ: A, З своей кампании

Формат запроса: Нет

Пример ответа: 200: `{"id":"77fd896bbd964b9cbc10f6323c39b4f8","status":"paused"}`

Коды ошибок: 404

GET `/campaigns/{campaign_id}`

Назначение: Состояние кампании и число контактов по статусам. Доступ: A, З своей кампании

Формат запроса: Нет

Пример ответа: 200: `{"id":"77fd896bbd964b9cbc10f6323c39b4f8","name":"Чат","status":"running","scenario_id":"demo_info","channel":"telegram","legal_basis":"service","contacts":{"pending":1}}`

Коды ошибок: 403, 404

GET `/campaigns`

Назначение: Перечень кампаний, заказчику только свои. Доступ: любой ключ

Формат запроса: Нет

Пример ответа: 200: `[{"id":"3ae88749b2b24703aa85fab22f058d01","name":"Голос","status":"draft","scenario_id":"demo_info"},{"id":"77fd896bbd964b9cbc10f6323c39b4f8","name":"Чат","status":"running","scenario_id":"demo_info"}]`

Коды ошибок: 401

Отчеты

GET `/reports/campaigns/{campaign_id}`

Назначение: Сводка кампании. Доступ: А, З своей кампании

Формат запроса: Нет

Пример ответа: 200: `{"campaign":"Чат","status":"running","contacts":1,"attempts":0,"reached":0,"reach_rate":0.0,"promised":0,"promised_sum_rub":0,"claims_paid":0,"dnc":0,"outcomes":{},"contacts_by_outcome":{},"talk_minutes":0.0,"cost_total_rub":null, ...`

Коды ошибок: 403, 404

GET `/reports/campaigns/{campaign_id}/xlsx`

Назначение: Сводка файлом. Доступ: А, З своей кампании

Формат запроса: Параметр `format`: xlsx (по умолчанию) или csv

Пример ответа: 200: книга XLSX 6 056 байт

Коды ошибок: 400 (неизвестный формат), 403, 404

GET `/reports/campaigns/{campaign_id}/csv`

Назначение: Детализация кампании CSV (разделитель точка с запятой, UTF-8 с BOM). Доступ: А, З своей кампании

Формат запроса: Нет

Пример ответа: 200: файл CSV 253 байт

Коды ошибок: 403, 404

GET `/reports/campaigns/{campaign_id}/losses`

Назначение: Воронка и причины потерь. Доступ: А, З своей кампании

Формат запроса: Нет

Пример ответа: 200: `{"campaign":"Чат","status":"running","funnel":{"contacts":1,"reached":0,"identity_confirmed":0,"informed":0,"result":0},"success":0,"in_progress":1,"in_progress_by_code":{"awaiting_attempts":1},"callbacks_scheduled":0,"lost":0,"lo ...`

Коды ошибок: 403, 404

GET `/reports/campaigns/{campaign_id}/ab`

Назначение: Итог сравнения редакций сценария. Доступ: А, З своей кампании

Формат запроса: Нет

Пример ответа: 200: `{"campaign":"Чат","ab":false,"note":"сравнение не настроено: кампания идет одним сценарием"}`

Коды ошибок: 403, 404

GET `/reports/campaigns/{campaign_id}/losses.xlsx`

Назначение: Потери файлом. Доступ: А, З своей кампании

Формат запроса: Нет

Пример ответа: 200: книга XLSX 6 371 байт

Коды ошибок: 403, 404

GET `/reports/campaigns/{campaign_id}/executive`

Назначение: Отчет руководителю, одна печатная страница. Доступ: A, З своей кампании

Формат запроса: Нет

Пример ответа: 200: страница HTML 3 728 байт

Коды ошибок: 403, 404

GET `/reports/attempts/{attempt_id}/proof`

Назначение: Акт по попытке: по звонку о направлении уведомления, по чату о приеме обращения. Доступ: любой ключ, заказчик только свои

Формат запроса: Нет

Пример ответа: 200: `АКТ О ПРИЕМЕ ОБРАЩЕНИЯ В ЧАТЕ Исполнитель: УК Пример Канал: web Абонент: web:cc2dd2deb34548f2a84ca6091a304fcb (веб - id сессии; мессенджер - id пользователя); лицевой счет: не назван Обращение начато: 28.09.2026 15:57 (Asia/Novosi ...`

Коды ошибок: 403, 404

GET `/reports/customers/{customer_id}/weekly`

Назначение: Письмо недели клиенту. Доступ: A, З

Формат запроса: Параметры `until` (дата), `days` 1-92, по умолчанию 7

Пример ответа: 200: страница HTML 4 667 байт

Коды ошибок: 404, 422

GET `/reports/customers/{customer_id}/appeals.xlsx`

Назначение: Журнал обращений за период. Доступ: A, З

Формат запроса: Параметры `until`, `days` 1-92

Пример ответа: 200: книга XLSX 7 798 байт

Коды ошибок: 404, 422

GET `/reports/customers/{customer_id}/flow`

Назначение: Поток обращений в чате за период. Доступ: A, З

Формат запроса: Параметры `until`, `days` 1-92 (по умолчанию 14), `work_from` 0-23, `work_to` 1-24

Пример ответа: 200: страница HTML 8 984 байт

Коды ошибок: 404, 422

GET `/reports/customers/{customer_id}/flow.xlsx`

Назначение: Поток обращений файлом. Доступ: A, З

Формат запроса: Те же параметры

Пример ответа: 200: книга XLSX 8 109 байт

Коды ошибок: 404, 422

Речевая аналитика (доступ: любой ключ, заказчик только свое)

POST `/analytics`

Назначение: Поставить пакет записей на разбор

Формат запроса: Форма: `customer_id`, `name` обязательны, `rules_id` (по умолчанию default), `source` upload или campaign, `campaign_id`, `attempt_ids` через запятую, `channels_role`, `language`, `retention_days` 1-365, `file` ZIP, CSV или XLSX

Пример ответа: 200: `{"id":"238d5a83cd20449f9ffd42e72213a627","status":"queued","queued":false,"note":"очередь недоступна: пакет принят и будет разобран позже"}`

Коды ошибок: 400 (нет файла, нет правил, неверный срок), 403, 404, 413, 503 (нет ffmpeg)

GET `/analytics/batches`

Назначение: Перечень пакетов

Формат запроса: Параметры `customer_id`, `limit` до 200, `offset`

Пример ответа: 200: `{"items":[{"id":"6b631e86bb194c3cadd9364a47dc5722","name":"Пакет","source":"upload","rules_id":"default","status":"queued","total":0,"done":0,"failed":0,"error":null,"created_at":"2026-09-28T08:57:18"},{"id":"238d5a83cd20449f9ffd4 ...`

Коды ошибок: 401

GET `/analytics/calls`

Назначение: Перечень записей

Формат запроса: Параметры `customer_id`, `batch_id`, `q`, `flagged`, `date_from`, `date_to` (ГГГГ-ММ-ДД), `limit`, `offset`

Пример ответа: 200: `{"total":0,"query":null,"items":[]}`

Коды ошибок: 400 (неверная дата)

GET `/analytics/calls/{call_id}`

Назначение: Запись с расшифровкой

Формат запроса: Параметр `q`

Пример ответа: Успешный ответ в прогоне не получен: разбор записей выполняет обработчик кампаний, в прогоне он не работал. Ответ: 404: `{"detail":"Not Found"}`

Коды ошибок: 403, 404

GET `/analytics/batches/{batch_id}`

Назначение: Сводка пакета, срок хранения и дата удаления

Формат запроса: Нет

Пример ответа: 200: `{"batch":"Пакет","status":"queued","rules":"default","calls":0,"done":0,"failed":0,"minutes":0.0,"recognised_seconds":0,"single_channel_calls":0,"avg_score":null,"avg_silence_ratio":null,"avg_script_compliance":null,"flags":{},"st ...`

Коды ошибок: 403, 404

DELETE `/analytics/batches/{batch_id}`

Назначение: Удалить записи и расшифровки пакета по акту, не дожидаясь срока. Повтор возвращает тот же акт

Формат запроса: Нет

Пример ответа: 200: `{"deleted":true,"act":{"batch_id":"238d5a83cd20449f9ffd42e72213a627","customer_id":"2d017d916c8a403c8d5f110281501855","deleted_at":"2026-09-28T08:57:18+00:00","by":"customer:2d017d916c8a403c8d5f110281501855","calls_purged":0,"segm ...`

Коды ошибок: 403, 404

GET `/analytics/report.xlsx`

Назначение: Отчет по пакету

Формат запроса: Параметр `batch_id`

Пример ответа: 200: книга XLSX 6 673 байт

Коды ошибок: 403, 404

Обработка документов (доступ: любой ключ, модуль лицензируется отдельно)

POST `/documents/recognize`

Назначение: Распознать текст документа (PDF, JPEG, PNG, TIFF, WebP)

Формат запроса: Форма: `file`, `customer_id`, `lang`, `pages` (вида 1, 1-3, 1,4-6), `psm` 0-13, `store`

Пример ответа: Успешный ответ в прогоне не получен: программа tesseract не установлена. Ответ: 503: `{"detail":"распознавание недоступно: tesseract не найден: tesseract"}`

Коды ошибок: 400, 403 (модуль не в лицензии), 413, 422, 503, 504

POST `/documents/parse`

Назначение: Извлечь реквизиты по шаблону из файла или по ссылке `ref`

Формат запроса: Форма: `file` или `ref`, `customer_id`, `template_id`, `lang`

Пример ответа: Успешный ответ в прогоне не получен: программа tesseract не установлена, шаблонов нет. Ответ: 503: `{"detail":"распознавание недоступно: tesseract не найден: tesseract"}`

Коды ошибок: 400, 403, 404, 413, 422, 503, 504

GET `/documents/templates`

Назначение: Перечень шаблонов документов

Формат запроса: Параметр `id`

Пример ответа: 200: `[]`

Коды ошибок: 403, 404

Вебхуки и веб-виджет (ключ не нужен, проверяются секреты каналов)

POST `/webhooks/max`

Назначение: Обновления MAX

Формат запроса: Заголовок `X-Max-Bot-Api-Secret`, тело обновления MAX

Пример ответа: 200: `{"ok":"true"}`

Коды ошибок: 401

POST `/webhooks/max/{customer_id}`

Назначение: Обновления MAX для заказчика

Формат запроса: То же

Пример ответа: 200: `{"ok":"true"}`

Коды ошибок: 401, 404

POST `/webhooks/telegram`

Назначение: Обновления Telegram

Формат запроса: Заголовок `X-Telegram-Bot-Api-Secret-Token`

Пример ответа: Успешный ответ по этому адресу не получен: секрет по умолчанию `TELEGRAM_WEBHOOK_SECRET` не задавался. Ответ: 401: `{"detail":"invalid webhook secret"}`

Коды ошибок: 401

POST `/webhooks/telegram/{customer_id}`

Назначение: Обновления Telegram для заказчика

Формат запроса: То же

Пример ответа: 200: `{"ok":"true"}`

Коды ошибок: 401, 404

POST `/webhooks/vk`

Назначение: События Callback API VK, ответ текстом

Формат запроса: JSON события с `group_id` и `secret`

Пример ответа: 200: `ok`

Коды ошибок: 401, 403

POST `/webhooks/vk/{customer_id}`

Назначение: События VK для заказчика. На событие confirmation возвращается код подтверждения

Формат запроса: То же

Пример ответа: 200: `abc123`

Коды ошибок: 401, 403, 404

POST `/webhooks/web/session`

Назначение: Выдать сеанс веб-виджета

Формат запроса: Нет

Пример ответа: 200: `{"session_id":"555aa13e3be64065b6de6be536d6ed82","token":"8bec64f09b579f9d4d18f4f3ac639f5dc95373401790f092a035c8e0cdcb7e69"}`

Коды ошибок: Нет

POST `/webhooks/web/{customer_id}/session`

Назначение: Выдать сеанс веб-виджета заказчика

Формат запроса: Нет

Пример ответа: 200: `{"session_id":"8cbaed2a27b84e7d849c23dc7d18651e","token":"cf6ccd3392cb62f2791249391b34d3d97947c0f44cc6ac7cf7643e101832c19a"}`

Коды ошибок: 404

POST `/webhooks/web`

Назначение: Сообщение веб-виджета

Формат запроса: JSON: `session_id`, `token` (64 знака), `text` до 4000 знаков, `photo`

Пример ответа: 200: `{"text":"Здравствуйте,. Это тестовое сообщение. Вам удобно говорить?","buttons":["Да","Нет"],"done":false}`

Коды ошибок: 400, 403 (`invalid session token`), 404

POST `/webhooks/web/{customer_id}`

Назначение: Сообщение веб-виджета заказчика

Формат запроса: То же

Пример ответа: 200: `{"text":"Спасибо, до свидания.\n\nЗдравствуйте,. Это тестовое сообщение. Вам удобно говорить?","buttons":["Да","Нет"],"done":true}`

Коды ошибок: 400, 403, 404

GET `/webhooks/web/widget`

Назначение: Страница веб-виджета

Формат запроса: Нет

Пример ответа: 200: страница HTML 1 836 байт

Коды ошибок: Нет

GET `/webhooks/web/{customer_id}/widget`

Назначение: Страница веб-виджета заказчика

Формат запроса: Нет

Пример ответа: 200: страница HTML 1 836 байт

Коды ошибок: 404

6.5. Выполнение: командная строка

Вызов python -m siberia_dialog.cli <команда>. Без команды программа печатает подсказку и завершается с кодом 2:

```
python -m siberia_dialog.cli
```

Ниже приведен исторический вывод указанной команды. Вывод программы не вводится в оболочку.

```
usage: siberia-dialog [-h]
                      {chat,validate,warmup,analytics-rules,rotate-secrets,passport,smoke}
                      ...
siberia-dialog: error: the following arguments are required: cmd
```

validate

Аргументы: Файлы сценариев

Что делает: Проверить сценарии

Ответ программы: `OK scenarios/demo_info.yaml: 2 состояний, 1 фикс. фраз, basis=service` или `ERR nope.yaml: [Errno 2] No such file or directory: 'nope.yaml'`

Код возврата: 0 если все OK, иначе 1

chat

Аргументы: Файл сценария, слоты ключ=значение

Что делает: Диалог по сценарию в терминале. Пустая строка означает молчание, одна цифра нажатие клавиши. Выход по Ctrl+D или Ctrl+C

Ответ программы: Реплики робота и кнопки, в конце `- конец, исход: informed`

Код возврата: 0

warmup

Аргументы: Файл сценария

Что делает: Синтезировать неизменяемые реплики в кэш TTS_CACHE_DIR, обращается к службе синтеза

Ответ программы: `<id>: N фраз, синтезировано новых: K`. При пустом YANDEX_API_KEY в прогоне: `httpx.LocalProtocolError: Illegal header value b'Api-Key '`

Код возврата: 0, при ошибке 1

analytics-rules

Аргументы: Файлы правил

Что делает: Проверить правила речевой аналитики

Ответ программы: `OK analytics_rules/default.yaml: 0 обязательных, 0 групп стоп-слов` или `ERR ...`

Код возврата: 0 если все OK, иначе 1

rotate-secrets

Аргументы: Нет

Что делает: Перешифровать настройки заказчиков действующим SECRET_KEY, прежний ключ берется из SECRET_KEY_PREVIOUS

Ответ программы: `перешифровано заказчиков: 0; теперь SECRET_KEY_PREVIOUS можно убрать`

Код возврата: 0

passport

Аргументы: Файл сценария, -o файл.md

Что делает: Каркас паспорта сценария для сдачи-приемки

Ответ программы: `паспорт: passport.md` или текст Markdown

Код возврата: 0

smoke

Аргументы: --круг чат|голос, --посев база|реестр

Что делает: Самопроверка по звеньям от заказчика до реестра показаний на SQLite, без сети. Требует DATABASE_URL на SQLite и сценариев zhkh_debt и zhkh_readings (круг чат) или zhkh_readings_voice (круг голос) в SCENARIOS_DIR, в архив они не входят

Ответ программы: Без SQLite: `ОТКАЗ: смоук-тест стирает схему базы, а DATABASE_URL ведет не на SQLite ...`. Без сценариев в прогоне: `[Ошибка] первое красное звено: 3 - кампания создается: 400 {"detail":"scenario zhkh_debt not found"}`

Код возврата: 0 все звенья прошли, 2 красное звено или отказ

`cli_license verify <файл>`

Аргументы: Файл лицензии

Что делает: Проверить подпись и срок лицензии

Ответ программы: JSON лицензии с полем status, в прогоне для неподписанного файла `"status": "invalid"`

Код возврата: 0 только при valid, иначе 1

`cli_license issue`

Аргументы: --licensee, --inn, --until, --lines, --channels

Что делает: Выпустить лицензию закрытым ключом правообладателя. Эксплуатанту не нужна

Ответ программы: Подписанный JSON лицензии

Код возврата: 0

`healthcheck worker`

Аргументы: Переменная REDIS_URL

Что делает: Проверить живость обработчика кампаний

Ответ программы: Пусто при успехе, иначе причина в поток ошибок

Код возврата: 0 жив, 1 нет

`healthcheck voice`

Аргументы: Переменная AUDIOSOCKET_PORT

Что делает: Проверить прием AudioSocket голосовым обработчиком

Ответ программы: Пусто при успехе, иначе причина, например `ConnectionRefusedError(...)`

Код возврата: 0 жив, 1 нет

Команды cli_license и healthcheck вызываются как python -m siberia_dialog.cli_license и python -m siberia_dialog.healthcheck. Пример диалога (тестовый сценарий эксплуатанта, ответ абонента подан через канал ввода, поэтому само слово «да» в выводе не отображается):

```
printf 'да\n' | python -m siberia_dialog.cli chat scenarios/demo_info.yaml full_name="Иван Петрович"
```

Ниже приведен исторический вывод указанной команды. Вывод программы не вводится в оболочку.

```
[Программа] Здравствуйте, Иван Петрович. Это тестовое сообщение. Вам удобно говорить?
   Да | Нет
[Пользователь] [Программа] Спасибо, до свидания.
- конец, исход: informed
```

6.6. Завершение

Новые вызовы набирает только обработчик кампаний и только по голосовым кампаниям в статусе running. Шаги лестницы касаний выполняются только у кампаний в статусе running. Перед плановой остановкой приостановить кампании (POST /campaigns/{campaign_id}/pause), дождаться, пока в ответе GET /campaigns/{campaign_id} не останется контактов calling, затем остановить голосовой обработчик, обработчик кампаний и сервер программного интерфейса.

Процессы программы завершаются по сигналу SIGINT (Ctrl+C) или SIGTERM, в том числе когда сигнал посылает средство, которым запущены службы. Сервер программного интерфейса при остановке закрывает подключение к очереди и общий HTTP-клиент чат-каналов. Журнал корректной остановки

(прогон 28.09.2026, остановка вызвана тем же обработчиком, что у сигнала SIGTERM):

```
INFO:     Shutting down
INFO:     Waiting for application shutdown.
INFO:     Application shutdown complete.
INFO:     Finished server process [6832]
```

Обработчик кампаний при остановке закрывает HTTP-клиент чат-каналов. Пакеты аналитики в статусе queued после запуска ставятся в очередь повторно каждые 5 минут. Поведение задачи, выполняющейся в момент остановки, определяется библиотекой очереди arq и в прогоне не проверялось.

Голосовой обработчик собственного обработчика сигналов не имеет. Разговоры, идущие в момент остановки, прерываются, их контакты остаются в статусе calling. После запуска обработчик кампаний не раньше чем через CALL_WATCHDOG_SEC (90 секунд) от начала вызова спрашивает Asterisk, существует ли канал. Если канала нет, попытка записывается с исходом no_answer, и контакт вне лестницы касаний получает повтор не раньше чем через 120 минут, пока число попыток меньше max_attempts. Если Asterisk не ответил, решение откладывается до следующего прохода. Перед записью попытки программа проверяет, не записана ли она уже по тому же каналу Asterisk. Этот путь в прогоне не проверялся, он описан по исходному тексту.

Команды командной строки завершаются сами с кодами из п. 6.5, команда chat по окончании сценария или по Ctrl+D, Ctrl+C.

6.7. Сообщения программы и действия администратора

`база часовых поясов недоступна: zoneinfo не видит ни одной зоны. ...`

Где: Запуск сервера, запуск прерывается

Что делать: Установить пакет tzdata

`нет языковых пакетов распознавания: ...,распознавание документов недоступно: ...`

Где: Запуск сервера, предупреждение

Что делать: Установить tesseract и пакеты rus, eng или задать TESSERACT_CMD

`языковая модель включена, обращения уходят на ...`

Где: Запуск сервера, предупреждение

Что делать: Убедиться, что адрес ведет к российской службе или к модели в контуре

`rate limit backend error: ...`

Где: Сервер

Что делать: Redis недоступен при RATE_LIMIT_BACKEND=redis, лимит не применяется. Восстановить Redis

`audit write failed: ...`

Где: Сервер

Что делать: Запись журнала аудита не удалась. Проверить базу данных

`очередь недоступна, задача ... отложена: ...`

Где: Сервер

Что делать: Redis недоступен, пакет будет поставлен в очередь повторно. Восстановить Redis

`речевая аналитика недоступна: ...`

Где: Запуск обработчика кампаний

Что делать: Установить ffmpeg и ffprobe

`свободных линий нет (предел: ...) - тик не набирает`

Где: Обработчик кампаний

Что делать: Дождаться освобождения линий или расширить лицензию, проверить TRUNK_CHANNELS

`campaign ...: обзвон остановлен - ...`

Где: Обработчик кампаний

Что делать: Устранить названную причину (маркировка, реестр отказов), набор возобновится сам

`сторож: не спросили Asterisk о канале ...`

Где: Обработчик кампаний

Что делать: Проверить доступность Asterisk ARI

`AudioSocket ... не поднялся (...)`

Где: Голосовой обработчик

Что делать: Исправить AUDIOSOCKET_HOST или освободить порт, процесс повторит попытку через 30 секунд

`входящий на ...: линия не настроена (voice_in-кампаний: N) - отбой`

Где: Голосовой обработчик

Что делать: Создать и запустить кампанию voice_in с номером в defaults.did

`ARI: обработчик события ... упал: ...`

Где: Голосовой обработчик

Что делать: Сбой одного события, прием продолжается. Разобрать по трассе в журнале

Описанные выше ограничения исторического прогона 28 сентября 2026 года сохраняются. Новые наблюдения приведены в разделе 8.

7. Резервное копирование и обновление

Резервному копированию подлежат база данных, каталог записей разговоров и файл лицензии. Обновление выполняется заменой экземпляра на архив новой версии с последующим применением миграций схемы базы данных. Принудительное обновление программой не инициируется.

8. Проверка редакции от 29 октября 2026 года

Использована отдельная рабочая копия точного архива и отдельное виртуальное окружение Python 3.12.14 на macOS. Установлены объявленные основные зависимости и драйвер aiosqlite. Общие пакеты из среды интерпретатора были доступны окружению. Состав фактических версий сохранен в протоколе, поэтому эта проверка не объявляется установкой на чистой операционной системе.

Подтверждено совпадение всех 161 записи исходного архива с распакованными файлами. При установке добавлен только пустой README.md. Сценарий demo_info подготовлен отдельно, архив программы не пересобирался.

Проверены: валидация сценария, завершенный диалог в консоли, запуск HTTP-службы, /health и /about, получение описания API, отказ без ключа, создание синтетического заказчика и кампании, загрузка контакта, запуск и пауза текстовой кампании, отчет JSON и открытие выгрузки XLSX, диалог веб-виджета и остановка HTTP-службы сигналом SIGINT.

Звонки, внешние мессенджеры, речевые провайдеры и языковые модели не вызывались. Службы PostgreSQL, Redis, Asterisk, OCR и голосовой обработчик в новой проверке не испытывались. Linux, остановка под нагрузкой и промышленное восстановление также не проверены.

Проверены четыре группы: архив и импорт, сценарий и консольный диалог, основные операции HTTP API, запуск и остановка службы. Все четыре группы прошли заданные технические критерии. Это не означает отсутствие функциональных замечаний: повторное приветствие виджета описано ниже. Полная проверка всех 73 операций не выполнялась. Подробные результаты и версии зависимостей оформлены отдельным протоколом. Ограничения раздела 6.7 сохраняются.

Наблюдение при завершении веб-диалога. В минимальном сценарии после ответа «да» виджет вернул done=true. Поле text при этом содержало прощание и повтор начального приветствия. Это расхождение сохранено отдельно от проверки доступности и формата API. Консольный диалог завершился одной завершающей репликой. Причина дополнительного приветствия в этой редакции не устанавливалась, исходный архив не исправлялся.

Новый минимальный сценарий предназначен только для проверки работы. В нем предусмотрено значение full_name по умолчанию для консольного запуска и веб-виджета. Сценарии эксплуатации разрабатываются и проверяются отдельно.