HUBTALK

Hubtalk Web API Инструкция

Как ваша система сама запускает обзвоны и получает результаты: принципы работы и технический справочник.

Актуальная версия всегда на сайте: docs.hubtalk.ai/ru/v4/web-api · 17 разделов.

Содержание

Как это работает

Web API

Ключ доступа и подготовка

Как проходит обзвон

Согласие, реестр и зона ответственности

Когда что-то пошло не так

Технический справочник

Обращение к API

Создание кампании

Пример: перезвоны по исходам

Объект кампании, блокеры и предупреждения

Поля кампании

Управление кампанией

Отчёт по загрузке номеров

Отдельные звонки

Звонок с сайта: Web SDK

Исходы звонка: `call_status` и `status_reason`

Вебхуки

Коды ошибок и пример от начала до конца

КАК ЭТО РАБОТАЕТ

Web API

Как ваша система сама запускает обзвоны: с чего начать и где что описано.

Этот раздел объясняет, что умеет Web API, как получить ключ доступа, как через него запустить обзвон и как читать ответы.

Весь раздел одним файлом — «Hubtalk Web API». Файл обновляется вместе с инструкцией: удобно передать разработчику или загрузить в ИИ-ассистента.

⬇ Скачать в MD ⬇ Скачать в PDF

Раздел состоит из двух частей, и они для разных людей:

- **«Как это работает»** не требует знания программирования: что происходит, кто за что отвечает и что нужно решить до первого запроса.
- **«Технический справочник»** — для разработчика: адреса запросов, поля, форматы ответов, коды ошибок и примеры.

> **Как читать контракт.** Всё, что названо в таблице, стабильно, если в таблице не сказано иное. Две вещи в контракт **не входят**: `error.message` (английское диагностическое пояснение) и поле `message` внутри элементов `blockers` / `warnings` в успешных ответах по кампаниям. Новые поля и новые коды со временем добавляются — незнакомое поле считайте непрозрачным, а незнакомый код разбирайте по `error.type` + `error.status`.

Что это такое и зачем

Обычно кампанию обзвона заводит человек: заходит в кабинет, создаёт кампанию, загружает файл с номерами, нажимает **Запустить**. Это удобно, пока обзвоны эпизодические.

Web API нужен, когда обзвон должен запускаться **сам** — из вашей системы, в тот момент, когда там произошло событие. Появилась просрочка по договору, закрылась заявка, наступила дата напоминания — ваша система отправляет платформе один запрос, и обзвон начинается. Человек в этот момент не участвует.

> **Главное свойство: под капотом это тот же продукт.** Кампания, созданная запросом, ничем не отличается от созданной в кабинете. Те же проверки перед запуском, тот же разбор номеров, тот же реестр «не звонить», тот же планировщик, те же ограничения частоты звонков на номер, те же записи разговоров и отчёты. Она видна в кабинете в общем списке, её можно поставить на паузу мышкой. Это не «второй продукт для разработчиков» — это вторая дверь в один и тот же.

Две модели работы — выберите одну

	Отдельные звонки	**Кампании**
Что вы отправляете	один номер за раз	список номеров (и потом ещё партии, если нужно)
Запрос	`POST /v1/calls/phone`	`POST /v1/campaigns`
Кто держит темп	вы	платформа
Кто повторяет недозвоны	вы	платформа, **если вы разрешили повторы**
Кто следит за рабочими часами	вы	платформа
Кто сверяет с «не звонить»	платформа	платформа
Когда это удобно	ваша система уже умеет быть дозвонщиком	вы хотите отдать список и получить результат

Смешивать их в одной задаче не нужно: если вы отдаёте список кампанией, платформа сама решает, кому и когда звонить, и вмешательство «сверху» отдельными звонками ломает её расчёт темпа.

Дальше раздел в основном про **кампании** — это то, ради чего Web API обычно и подключают; отдельные звонки описаны в Отдельных звонках.

Как это работает

- Ключ доступа и подготовка — Как выпустить ключ, что должно быть готово до первого запроса и чек-лист перед запуском.
- Как проходит обзвон — Что происходит после запроса, как читать ответ и зачем обязателен ключ повтора.
- Согласие и ответственность — Подтверждение согласия, реестр «не звонить» и что платформа делает сама.
- Когда что-то пошло не так — Частые отказы, что они означают и что с ними делать.

Технический справочник

- Обращение к API — Базовый адрес, аутентификация, форматы и карта всех запросов.
- Создание кампании — `POST /v1/campaigns`: заголовки, тело запроса, порядок отказов.
- Объект кампании — Что приходит в ответе и как читать блокеры и предупреждения запуска.
- Поля кампании — Повторы, номера, привязка набора, согласие, сверка с реестром, идемпотентность.
- Управление кампанией — Запуск, пауза, отмена, прогресс, статистика и добавление партий номеров.
- Отчёт по загрузке номеров — Что вернулось по каждой строке файла и построчные коды замечаний.
- Отдельные звонки — `POST /v1/calls/phone`, объект звонка, чтение, завершение и список.
- Исходы звонка — Ось исходов: `call_status`, `status_reason` и как они складываются в вердикт.
- Вебхуки — События `call_ended` и `call_analyzed`, подпись, повторы и порядок доставки.
- Коды ошибок и пример — Полный перечень кодов и рабочий сценарий одним куском.

КАК ЭТО РАБОТАЕТ

Ключ доступа и подготовка

Как выпустить ключ, что должно быть готово до первого запроса и чек-лист перед запуском.

Ключ доступа: как получить и что с ним делать

Ключ — это длинная строка, которой ваша система представляется платформе. Он заменяет логин и пароль: у программы нет человека, который вводил бы их руками.

Как выпустить

1. Зайдите в кабинет под учётной записью **администратора или разработчика** вашей организации.
2. Откройте раздел **«API-ключи»** в левом меню — это отдельный пункт меню, а не подраздел «Настроек».
3. Нажмите **«Создать ключ»**.
4. Заполните:
   - **Имя** — для вас, чтобы потом понять, чей это ключ («Биллинг», «CRM продакшен»). В запросах оно не участвует.
   - **Лимит запросов в минуту** — сколько обращений в минуту разрешено этому ключу. Пусто или `0` — общий лимит платформы (обычно 60 в минуту; точное действующее значение показано рядом с полем). Ограничение защищает и вас: сломавшаяся интеграция не положит очередь.
5. Нажмите **«Создать»**.

> **Секрет показывается ровно один раз.** В окне появится строка вида `cfk_...`. Скопируйте её и передайте разработчику по защищённому каналу — платформа хранит только хеш и повторно показать секрет не сможет. Если секрет потерян, ключ отзывают и выпускают новый.

Что ещё можно

- **Изменить лимит** запросов — карандашом в списке. Секрет при этом не меняется, интеграцию перенастраивать не нужно; новый лимит действует со следующего запроса.
- **Отозвать ключ** — строка остаётся в списке с пометкой «отозван» (это нужно для аудита), но запросы с ним сразу перестают работать.

Чего ключ не может

- **Выйти за пределы вашей организации.** Ключ видит только её кампании, агентов, линии и загрузки. Всё, что за этой границей, для него не «запрещено» — его для него просто не существует: ответ такой же, как на несуществующий объект.
- **Заменить настройку телефонии.** Ключ не создаёт линии и не привязывает агентов к ним: это делает администратор в кабинете.

Почему важно, КТО выпустил ключ

Платформа записывает, какой сотрудник и в какой момент выпустил ключ. Это не формальность аудита, а юридическое основание: запуск обзвона требует подписи под подтверждением, что у вас есть согласие абонентов, а подпись под ним ставит ключ — то есть программа. Действие программы имеет силу до тех пор, пока его можно связать с человеком, который взял на себя обязательство. Этот человек — тот, кто выпустил ключ. Запись «кто и когда» хранится в самой подписи и не собирается задним числом.

Практический вывод: **выпуск ключа — не техническая мелочь.** Выпускать его должен тот, кто вправе отвечать за обзвоны от лица организации.

Что должно быть готово до первого запроса

Перечисленное настраивается в кабинете один раз; без этого обзвон не начнётся, и платформа честно скажет, чего не хватает.

Что	Кто делает	Зачем
Транк с исходящей группой	администратор в разделе «Телефония»	`trunk_id` и `route_id` группы. Пул номеров, разрешённые направления и лимиты звонков принадлежат **группе**, а не транку
Агент опубликован	ваш разработчик сценариев	звонит именно опубликованная версия; черновик не набирается
Агент привязан к исходящей группе	администратор	иначе звонить не с чего: платформа не выбирает линию за вас
Пакет рынка (для США)	поддержка платформы	без него нельзя проверить географию и законное окно звонков
Окно обзвона организации	администратор платформы	за пределами этого окна платформа не звонит, какие бы часы ни назвала кампания
Ключ выпущен	администратор или разработчик, раздел «API-ключи»	секрет `cfk_…`, см. Ключ доступа
Зарегистрирован адрес для вебхуков	администратор, на агенте: редактор → вкладка **«Настройки»** → **«Вебхуки (после звонка)»**	ваш URL и секрет подписи, переданный вам отдельным каналом

Отдельно про привязку: если агент привязан ровно к одной исходящей группе, **ничего указывать не нужно** — платформа найдёт её сама по агенту. Если групп несколько, она не станет выбирать за вас и попросит назвать линию явно: догадка здесь означала бы звонки с чужого номера и по чужой квоте, а узнали бы вы об этом по счёту.

Чек-лист перед первым запуском

1. Агент создан и **опубликован**; тестовый звонок из кабинета проходит.
2. Агент привязан к исходящей группе; если групп несколько — вы знаете, какую называть.
3. Для США: пакет рынка установлен, в списке номеров есть колонки географии.
4. Окно обзвона организации задано так, как вам нужно; рабочие часы кампании его не расширят.
5. Ключ выпущен **тем, кто вправе отвечать за обзвоны**, секрет передан разработчику по защищённому каналу и сохранён в вашем хранилище секретов.
6. Лимит запросов у ключа выставлен осознанно.
7. Ваша система формирует **новый** ключ повтора на каждую новую кампанию и каждую партию.
8. Цель обзвона (`classification`) задана явно и соответствует тому, о чём агент действительно говорит.
9. Решено, кто перезванивает недозвоны — вы или платформа, — и число попыток задано соответственно (по умолчанию — одна).
10. Первая кампания запущена на **коротком списке своих номеров**, ответ прочитан целиком, отчёт по строкам просмотрен.
11. Проверено, что кампания видна в кабинете и её можно поставить на паузу мышкой — это подтверждает, что интеграция работает с тем же продуктом.
12. Разобрано, кто у вас читает прогресс и что делает с блокерами, если обзвон не начался ночью и никто этого не увидел.

Дальше — для разработчика: как обращаться к API, какой запрос что делает и что приходит в ответ.

Изменения контракта от 14 октября 2026 — проверьте свой разбор ответов

- **В транскрипте только то, что абонент услышал.** Реплики агента, которые так и не прозвучали, больше не входят в `transcript` — в вебхуках `call_ended` и `call_analyzed`, в `GET /v1/calls/{id}` и в колонках транскрипта отчётов. Пометки `pending` и `unspoken` больше не приходят. Так же и для старых звонков при повторной отправке вебхука. См. объект звонка.
- **Две новые причины** `status_reason`: `caller_abandoned` (`not_connected`) — звонящий на входящий положил трубку до ответа агента, раньше это был `no_speech`; `no_media` (`error`) — абонент ответил, но звука от него не было, если у агента не выключено это определение (по умолчанию оно включено). См. статусы звонка.
- **Скорость обзвона — любое целое от 1 до 200**, а не одна из ступеней, и теперь её можно поменять у созданной кампании через `PATCH /v1/campaigns/{id}`.

Номер версии не менялся. Значение `status_reason`, которое ваш разбор не знает, обрабатывайте по его корзине `call_status`.

Три изменения контракта от 22 сентября 2026 — проверьте свой разбор ответов

Вместе с отменой общего потолка набора (см. скорость обзвона) из ответов ушли три поля. Ни одно из них не заменено другим — их просто больше нет, потому что нет и механизма, о котором они сообщали:

- в `warnings[]` при создании кампании кода `rate_clamped` нет;
- в ответе `GET /v1/campaigns/{id}/progress` поля `rate.clamped` нет;
- в списке лимитов кампании поле `max_dials_per_minute` заменено списком доступных ступеней `calls_per_minute_tiers`.

Разбор, который читает `rate.clamped` без проверки на наличие ключа, на этом упадёт. Значение `calls_per_minute` всегда равно тому, что вы прислали (или задали позже через `PATCH`).

КАК ЭТО РАБОТАЕТ

Как проходит обзвон

Что происходит после запроса, как читать ответ и зачем обязателен ключ повтора.

Как проходит обзвон, запущенный запросом

Шаг 1. Ваша система отправляет один запрос. В нём: название кампании, имя агента, скорость обзвона, рабочие часы, цель обзвона, подтверждение согласия и сам список номеров с данными для разговора.

Шаг 2. Платформа разбирает список. Тот же разбор, что у файла в кабинете: нормализация номеров, поиск дублей, проверка обязательных колонок, сверка с реестром «не звонить», проверка географии для США. Ничего из этого на пути через API не пропускается и не упрощается.

Шаг 3. Платформа проверяет, можно ли запускать. Опубликован ли агент, есть ли линия, не в прошлом ли окно, подписано ли подтверждение, не все ли номера оказались в реестре «не звонить».

Шаг 4. Если всё в порядке — обзвон начинается. Тем же планировщиком, что и у кампаний из кабинета: с вашей скоростью, в ваши рабочие часы, с повторами недозвонов по вашим правилам и с учётом ограничений частоты звонков на один номер, действующих в вашей организации.

Шаг 5. Вы читаете ответ. В нём — что получилось: поехал ли обзвон, сколько строк принято, что не так с непринятыми.

Отдельного «нажатия кнопки» в этой схеме нет: один запрос делает все три вещи сразу — создаёт, принимает номера и запускает.

Как читать ответ

Ответ приходит **всегда**, даже когда обзвон не начался. Читается он по одному полю и двум спискам.

`status` — единственный ответ на вопрос «поехало ли»:

Значение	Что означает
`running`	обзвон идёт
`scheduled`	всё в порядке, ждём начала окна
`draft`	не начался, причина — в списке `blockers`

`blockers` — почему не начался. Это не «ошибки запроса», а состояние дел: агент снят с публикации, номера попали в реестр «не звонить», окно в прошлом, нет подписи под подтверждением согласия. Каждая причина названа отдельным кодом — тем же, что виден в кабинете на экране запуска.

> Одна тонкость, о которой стоит знать заранее: если вы **позже читаете карточку кампании** отдельным запросом, `blockers` там отвечает на вопрос «что сказала бы проверка запуска прямо сейчас», а не «почему кампания не работает». У работающей кампании этот список никогда не пуст — как минимум в нём стоит «нужна подпись подтверждения», потому что подпись даётся на каждый запуск, а не хранится как состояние. Решает вопрос «идёт ли обзвон» только поле `status`.

`warnings` — предупреждения. Они ничего не блокируют и есть почти у каждой кампании: например, «начало не задано — набор начнётся сразу» и «действует частотная политика» с цифрами, сколько раз в сутки и за неделю можно звонить на один номер.

`import` — что стало со списком номеров. Сколько строк увидели, сколько приняли, сколько с предупреждениями, сколько отклонили, сколько дублей, и по каким причинам (сгруппировано по коду причины с номерами первых строк). `dnc_screen` отдельно называет: сколько читаемых строк проверено, сколько не

проверено, сколько строк совпало с реестром и сколько это уникальных номеров; каждое совпадение несёт строку, исходный телефон, имя, причину и запись реестра.

> **Кампания и отчёт сохраняются в любом исходе.** Даже когда обзвон не начался, кампания остаётся в кабинете, а отчёт по строкам доступен: это единственный способ понять, что не так с данными, и удалять его вместе с неудачей значило бы уничтожить ровно то, что нужно для исправления.

Одно правило, которое избавляет от догадок

Действие, которое ничего не изменило, — всегда ошибка. Успешный ответ означает, что состояние в теле ответа и есть новое состояние. Поэтому:

- запрос на создание отвечает успехом, даже если обзвон не начался: кампания и отчёт действительно созданы, и `status` честно говорит `draft`;
- отдельный запрос «запусти», который ничего не запустил, отвечает **ошибкой** — и называет причину в самой ошибке, чтобы не пришлось спрашивать второй раз;
- повторная команда «поставь на паузу» уже остановленной кампании — **не** ошибка: состояние и так такое, как вы просили.

Повторная отправка: почему обязателен «ключ повтора»

Сеть иногда обрывает ответ уже после того, как платформа всё сделала. Ваша система в этот момент не знает, дошёл запрос или нет, и обычно повторяет его. Для одного звонка это неприятно; для списка из десяти тысяч номеров это означает **обзвонить всех дважды**.

Поэтому на создании кампании **и** на добавлении номеров в неё обязателен заголовок `Idempotency-Key` — произвольная строка, которую ваша система придумывает для каждой новой кампании или партии (например, идентификатор задачи в вашей системе). Дальше платформа разбирается сама:

- **тот же ключ, то же тело** — вернётся тот же ответ, что и в первый раз; вторая кампания не появится;
- **тот же ключ, пока первый запрос ещё выполняется** — отказ `409 idempotency_key_in_flight`: подождите ответ, не повторяйте снова;
- **тот же ключ, но ДРУГОЕ тело** — отказ `422 idempotency_key_reuse`. Это защита от самой неприятной ошибки: без неё вы получили бы ответ на чужой запрос и решили бы, что платформа проигнорировала ваши настройки;
- **запрос отклонён до создания кампании** — ключ освобождается сразу, и исправленный запрос с тем же ключом пройдёт нормально. Если отказ случился уже после создания (например, не удалось сохранить список), ключ остаётся занятым, а созданную кампанию можно найти в списке кампаний.

Ключи помнятся сутки. Практический вывод: **новая кампания — новый ключ, новая партия — новый ключ.** Повторять с прежним ключом стоит только буквально тот же запрос.

КАК ЭТО РАБОТАЕТ

Согласие, реестр и зона ответственности

Подтверждение согласия, реестр «не звонить» и что платформа делает сама, а что остаётся вам.

Подтверждение согласия и цель обзвона

Перед запуском платформа требует подтвердить, что у вас есть согласие абонентов на звонки. В кабинете это галочка с юридическим текстом; через API — поле `attestation` в запросе. Без него кампания создаётся, но остаётся черновиком с причиной «нужна подпись».

Подпись ставится **ключом**, а человеческое звено — сотрудник, выпустивший ключ (Ключ доступа). Подпись привязана к конкретной версии данных: подтверждение, данное под одним списком номеров, не переносится на следующий. Загрузили новый список или добавили партию номеров — подтверждаете заново.

Цель обзвона (`classification`) — чем именно вы звоните:

Значение	Про что	Тип согласия
`marketing`	предложения, реклама, опросы	`pewc` — письменное согласие (самый строгий)
`servicing`	обслуживание: напоминания, уведомления, статусы	`pec` — явное согласие
`debt_collection`	взыскание задолженности	`pec` — явное согласие

От цели зависит **тип согласия** (`consent_tier`) — какой именно юридический текст вы подписываете. Поэтому платформа её не угадывает: не указали — берётся `marketing`, самый строгий вариант. В ответе приходит и цель, и получившийся тип, чтобы вы видели, под каким текстом подписались.

Реестр «не звонить»

Номера из реестра вашей организации платформа находит **до** первого звонка и перечисляет поимённо. Проверяется каждая читаемая строка списка до удаления дублей: поэтому три строки одного номера — это три совпавшие строки, но один исключённый номер. Строки с нечитаемым телефоном считаются «непроверенными», а ноль совпадений всегда возвращается явным статусом `clean`, а не выводится из пустого списка. Запуск при этом останавливается.

Дальше решаете вы:

- **оставить блокировку** (`dnc: "block"` , значение по умолчанию) — совпавшие номера не набираются;
- **подтвердить обзвон** (`dnc: "skip"`) — платформа записывает это решение в аудит ровно так же, как кнопку в кабинете, с указанием, к какой загрузке **и к какому точному составу совпадений** оно относится. Появилась новая запись реестра по тому же списку — согласие сбрасывается, и проверка запуска снова показывает обновлённый список совпадений.

Если в реестре оказались **все набираемые номера**, подтверждать нечего — запуск останавливается безусловно. Непосредственно перед запуском платформа снова читает живой реестр и сохраняет его «снимок» вместе с подписью; после запуска карточка кампании и отчёт о совпадениях показывают именно этот снимок, даже если реестр потом менялся. Номера, попавшие в реестр уже по ходу кампании, всё равно не набираются — прогресс кампании считает их отдельно.

> Для партий номеров, добавляемых в уже созданную кампанию (Согласие и ответственность), правило жёстче: совпадение с реестром не «останавливает запуск», а **отклоняет всю партию** — подтвердить обзвон таких номеров через добавление нельзя.

Что платформа делает сама, а что остаётся вам

Делает платформа: разбор номеров и переменных, дубли, реестр «не звонить», география и законное окно звонков (США), окно обзвона организации, темп обзвона, повторы недозвонов, ограничение частоты звонков на номер, часовые пояса контактов, записи разговоров, разбор результатов.

Остаётся вам: решить, когда запускать кампанию и с каким списком; хранить у себя связь между вашими сущностями и кампанией; читать прогресс, если нужен статус в вашем интерфейсе; получать результаты звонков (вебхуки по каждому звонку работают так же, как для отдельных звонков).

Про повторы недозвонов. По умолчанию кампания делает **одну попытку** на номер и никого не перезванивает. Если хотите, чтобы платформа перезванивала сама, задайте число попыток и паузу между ними (`retry_config`). Если перезвонами управляет ваша система (повторно присылает номер как собственный перезвон), оставьте одну попытку — иначе перезвон получится двухуровневым: ваш плюс платформенный. Повторно присланный номер, по которому кампания уже звонила, принимается как новый звонок и набирается снова.

Про рабочие часы. Если вы не задали кампании рабочие часы, это не значит «звонить круглосуточно»: платформа звонит в пределах окна обзвона вашей организации (а в США — в пределах законного окна юрисдикции абонента). Часы кампании могут только **сужать** это окно; попытка задать более широкие часы отклоняется сразу.

Добавить номера в уже созданную кампанию — можно, отдельным запросом `POST /v1/campaigns/{id}/calls`: партия несёт номера и подпись, а агент, линия, рабочие часы и повторы берутся из кампании. Работает для любой незакрытой кампании — черновика, запланированной, идущей или приостановленной. Правила, о которых надо знать заранее:

- **партия принимается целиком или отклоняется целиком.** Одна нечитаемая строка — и не уйдёт ни один номер из партии (`calls_rejected`); один номер из реестра «не звонить» — тоже вся партия, но своим кодом (`calls_dnc_matched`), чтобы ваша система отличала «исправить данные» от «убрать этих людей из базы». Ответ несёт разбор отклонённой партии с номером загрузки и первыми строками; полный отчёт — как у любой загрузки;
- **партия принимается и тогда, когда кампания сейчас не звонит** — например, стоит на паузе. Ответ скажет, что номера приняты, и назовёт причину простоя (`campaign_paused`); набор продолжится после возобновления;
- **обычная кампания закрывается сама, когда очередь опустела** — и после этого номера в неё не принимаются вовсе. Чтобы кампания ждала номера, создайте её **открытой к добавлению звонков**: `accepts_call_additions: true` с обязательной датой окончания (не дальше 30 суток). Такую кампанию можно создать пустой, наполнять партиями и закрыть отменой либо дождаться даты. Признак задаётся при создании и потом не меняется;
- в одной партии — **не больше 1000 номеров**; больше — разбивайте.

Сознательно не делается — не ищите этих возможностей, их нет:

- **отдельное событие «кампания завершилась».** Состояние читается запросом прогресса или карточки кампании;
- **отдельная выдача результатов кампании с расшифровками.** Результаты читаются как обычные звонки;
- **разные права у разных ключей.** Ключ либо есть, либо нет; ограничение доступа по разделам не предусмотрено.

КАК ЭТО РАБОТАЕТ

Когда что-то пошло не так

Частые отказы, что они означают и что с ними делать.

Ошибки приходят машинным кодом и английским пояснением. **Ориентируйтесь на код:** текст пояснения может меняться. Вот те, что встречаются в жизни чаще всего; полный перечень — в Коды ошибок.

Код	Что случилось	Что делать
`idempotency_key_required`	не прислан ключ повтора	добавить заголовок, см. Как проходит обзвон
`idempotency_key_reuse`	тот же ключ с другим телом	взять новый ключ для новой кампании или партии
`idempotency_key_in_flight`	первый запрос с этим ключом ещё выполняется	дождаться его ответа, не повторять
`campaign_agent_unknown`	агента с таким именем в вашей организации нет	проверить имя агента; неопубликованный агент — другой случай, он даёт черновик, а не отказ
`dialing_route_unresolved`	агент не привязан ни к одной линии	привязать в разделе «Телефония»
`dialing_route_ambiguous`	агент привязан к нескольким	назвать линию явно, см. Привязку набора
`dialing_binding_incomplete`	назван номер или группа без линии	прислать ещё и линию, либо не присылать ничего из трёх
`campaign_timezone_unknown`	часовой пояс не распознан	название вида `Asia/Almaty`; смещения вида `+05:00` не принимаются
`campaign_classification_invalid`	цель обзвона не из списка	одно из трёх значений Согласие и цель обзвона
`campaign_window_invalid`	окончание раньше начала	проверить окно
`campaign_spec_invalid`	отклонены настройки: скорость вне диапазона 1–200, неверные повторы, рабочие часы, действие по дублям или длина названия	сверить настройки со справочником
`working_hours_wider_than_law`	рабочие часы шире окна обзвона организации или законного окна	сузить часы или не задавать их вовсе
`launch_blocked`	запуск не прошёл проверки	причины перечислены в самой ошибке; блокер по реестру несёт обновлённый список совпадений
`campaign_transition_invalid`	переход невозможен из текущего состояния (например, возобновить отменённую)	проверить состояние кампании
`market_package_not_installed`	для вашей организации не установлен пакет рынка, номера нельзя проверить	обратиться в поддержку
`dnc_screen_required`	загрузка старого формата, без построчной сверки с реестром	загрузить номера заново
`report_filter_invalid`	неверное значение `only` у отчёта по загрузке	убрать фильтр или передать `only=dnc_match`

Код	Что случилось	Что делать
`calls_rejected`	в партии добавляемых номеров есть строка с ошибкой данных — партия не принята целиком	исправить строки из разбора в ошибке и прислать партию заново с тем же ключом повтора
`calls_dnc_matched`	в партии есть номер из реестра «не звонить» — партия не принята целиком	убрать такие номера из своей базы (список — в отчёте загрузки `only=dnc_match`) и прислать заново
`campaign_closed`	кампания завершена или отменена и номера больше не принимает	создать новую кампанию
`calls_required` / `calls_batch_too_large`	партия пуста или больше 1000 строк	присылать от 1 до 1000 номеров за запрос
`attestation_invalid`	подпись партии не в форме `{"confirmed": true}`	прислать ровно это поле, без `import_id`
`attestation_import_id_required`	при отдельном запуске не назван номер загрузки, под которой ставится подпись	передать `import_id` из ответа на создание
`campaign_open_ends_at_required` / `campaign_open_ends_at_too_far`	у кампании, открытой к добавлению звонков, нет даты окончания или она дальше 30 суток	назвать дату окончания в пределах горизонта
`campaign_auto_speed_ends_at_required` / `campaign_auto_speed_working_hours_required`	авто-скорости нужна дата окончания, а кампании на несколько дней — ещё и рабочие часы	передать `ends_at` и `working_hours` или выключить `auto_speed`
`campaign_not_found` / `import_not_found`	такой кампании или загрузки в вашей организации нет	проверить идентификатор

ТЕХНИЧЕСКИЙ СПРАВОЧНИК

Обращение к API

Базовый адрес, аутентификация, формат и карта всех запросов.

Как обращаться к API

- **Базовый адрес:** `https://{{HUBTALK_FQDN}}` — все пути ниже начинаются с `/v1`. Вместо `{{HUBTALK_FQDN}}` подставьте адрес, на котором работает ваш кабинет: `app.hubtalk.kz` — Казахстан, `app.hubtalk.uz` — Узбекистан.
- **Формат:** JSON в запросе и в ответе. Время — **секунды эпохи**.
- **Машиночитаемая схема:** `GET https://{{HUBTALK_FQDN}}/openapi.json`, интерактивный обозреватель — `https://{{HUBTALK_FQDN}}/docs`. Схема авторитетна для **запросов**; успешные ответы опубликованы там нетипизированными объектами, поэтому контракт ответов — таблицы на этой странице.

Аутентификация

Каждый запрос `/v1` несёт ключ:

```
Authorization: Bearer YOUR_API_KEY
```

Секрет (`cfk_` + hex) выпускается в кабинете и показывается один раз (Ключ доступа); платформа хранит только SHA-256-хеш. Отозванный ключ перестаёт работать сразу — запись ключа перечитывается на каждом запросе. Ключ ограничен организацией, в которой выпущен: всё, что за этой границей, неотличимо от отсутствующего (`404`), а не `403`.

Отказ	Статус	`error.type`	`error.code`
нет или неверный заголовок `Authorization`	401	`authentication_error`	`api_key_required`
неизвестный ключ	401	`authentication_error`	`api_key_unknown`
отозванный ключ	401	`authentication_error`	`api_key_revoked`
превышен лимит запросов ключа	429	`rate_limit_error`	`api_key_rate_limit_exceeded` (`Retry-After: 60`)

Лимит на ключ. У каждого ключа скользящее окно 60 секунд по всем его запросам `/v1`. Лимит задаётся ключу при создании и может быть **изменён позже** в кабинете без ротации секрета (действует со следующего запроса). Лимит `0` означает общий лимит платформы — это настройка развёртывания, 60 запросов в минуту, если оператор не поменял; оператор может и отключить лимит. Это один из **четырёх источников 429**.

Один конверт ошибки

Все ошибки `/v1` выглядят одинаково:

```
{
  "error": {
    "type": "rate_limit_error",
    "code": "dial_queue_full",
    "status": 429,
    "message": "Dial queue is full; retry after the interval in Retry-After"
  }
}
```

HTTP-статус	`error.type`
400, 422	`invalid_request_error`

HTTP-статус	`error.type`
401	`authentication_error`
403	`permission_error`
404	`not_found_error`
409	`conflict_error`
429	`rate_limit_error`
503	`service_unavailable_error`
500, 502	`api_error`

`error.code` называет точную причину внутри типа и присутствует **всегда**: если более конкретная причина не зарегистрирована, `code` равен `type`. Новые коды со временем добавляются — незнакомый код разбирайте по `type` + `status`.

Два дополнительных поля приходят только на конкретных отказах по кампаниям и больше нигде: `error.blockers` на `409 launch_blocked` и `error.import` на `422 calls_rejected` / `422 calls_dnc_matched`. Любая другая ошибка несёт ровно четыре поля выше.

> Класс ошибки определяйте по `error.type` + `error.status`, точную причину — по `error.code`. **Никогда не разбирайте** `error.message` — это человекочитаемое английское пояснение, подбираемое по коду на границе API, и частью контракта оно не является. На ошибках схемы `422` (`code: "invalid_request"`) `message` — **массив** замечаний по полям в форме FastAPI/pydantic (`{"type", "loc", "msg", "input"}`), а не строка.

Отказы проносят свои заголовки через конверт: `Retry-After` на 429 приходит всегда.

Лимиты и `429`

`429` приходит из четырёх независимых источников. Все несут `error.type: "rate_limit_error"` — различайте по `error.code`:

`error.code`	Источник	Что делать	`Retry-After`
`api_key_rate_limit_exceeded`	скользящее окно 60 секунд по всем запросам с ключом	снизить общий темп запросов	да, всегда `60`
`dial_queue_full`	очередь набора платформы упёрлась в потолок глубины; **звонок не создан**	повторить тот же запрос с **тем же** `Idempotency-Key` через указанный интервал	да (оценка очереди, 1–60 с)
`subgroup_concurrency_limit`	исчерпан лимит одновременных звонков группы	ждать завершения активных звонков; честной оценки не существует, используйте свой бэкофф	нет (осознанно)
`subgroup_rate_limit`	исчерпан лимит инициаций в минуту у группы	выждать минутное окно	да (`60`)

Лимиты группы по умолчанию — 10 одновременных звонков и 30 инициаций в минуту, если администратор не задал другие; для транка, используемого по гранту, действует меньшее из лимита группы и лимита гранта.

Карта запросов

Кампании

Метод и путь	Зачем
`POST /v1/campaigns`	создать кампанию, передать номера и запустить — один запрос

Метод и путь	Зачем
`GET /v1/campaigns`	список кампаний с курсорной постраничностью (краткие представления)
`GET /v1/campaigns/{id}`	текущее состояние кампании
`GET /v1/campaigns/{id}/launch-check`	`{blockers, warnings, dnc_screen}` до действия
`POST /v1/campaigns/{id}/launch`	запустить кампанию, которая осталась черновиком
`POST /v1/campaigns/{id}/calls`	добавить партию номеров в уже созданную кампанию
`POST /v1/campaigns/{id}/pause`	остановить набор новых номеров
`POST /v1/campaigns/{id}/resume`	продолжить (проверки запуска выполняются заново)
`POST /v1/campaigns/{id}/cancel`	отменить необратимо
`PATCH /v1/campaigns/{id}`	сменить скорость обзвона или включить авто-скорость
`GET /v1/campaigns/{id}/progress`	состояния контактов, удержания, темп, счётчики реестра и частоты
`GET /v1/campaigns/{id}/rate-history`	журнал пересчётов авто-скорости

Загрузки номеров

Метод и путь	Зачем
`GET /v1/imports/{import_id}`	сводка по загрузке плюс живая или зафиксированная сверка с реестром
`GET /v1/imports/{import_id}/report?format=csv`	отчёт по строкам, `csv` или `xlsx`
`GET /v1/imports/{import_id}/report?only=dnc_match`	только совпадения с реестром «не звонить», `csv` или `xlsx`

Линии

Метод и путь	Зачем
`GET /v1/dialing-routes?agent_id=…`	какие транки и группы доступны агенту

Отдельные звонки

Метод и путь	Зачем
`POST /v1/calls/phone`	один исходящий звонок
`POST /v1/calls/web`	web-звонок: сессия для браузера, см. «Звонок с сайта»
`GET /v1/calls`	список звонков с фильтрами и курсором
`GET /v1/calls/{call_id}`	полный объект звонка
`GET /v1/calls/{call_id}/turns`	метрики задержек по тёрнам завершённого звонка
`POST /v1/calls/{call_id}/end`	завершить активный звонок

ТЕХНИЧЕСКИЙ СПРАВОЧНИК

Создание кампании

`POST /v1/campaigns`: заголовки, тело запроса, порядок отказов и успешный ответ.

Один запрос делает три вещи: создаёт кампанию, принимает запланированные звонки и — если все проверки запуска прошли — начинает набор. Отдельного шага «старт» в счастливом пути нет.

Заголовки

Заголовок	Обязателен	Примечания
`Authorization: Bearer YOUR_API_KEY`	да	
`Content-Type: application/json`	да	
`Idempotency-Key: <уникальная строка>`	**да**	В отличие от `POST /v1/calls/*`, здесь не опционален: повтор после сетевого таймаута создал бы вторую кампанию и обзвонил весь список дважды. Отсутствует или пуст → `422 idempotency_key_required`. Протокол — в конце этого раздела, «Идемпотентность запросов по кампаниям».

Тело запроса

Неизвестные поля верхнего уровня игнорируются.

Поле	Тип	Обяз.	По умолч.	Описание
`name`	string	да	—	Название кампании, видное в кабинете; 1–80 символов после обрезки пробелов (иначе `422 campaign_spec_invalid`).
`agent_id`	string	да	—	**Имя** агента, разрешается внутри вашей организации. Неизвестное имя → `422 campaign_agent_unknown`. Существующий, но неопубликованный агент принимается: кампания создаётся черновиком с блокером `agent_not_published`.
`calls_per_minute`	integer	да, кроме `auto_speed: true`	—	Любое целое от 1 до 200 (вне диапазона → `422 campaign_spec_invalid`; нецелое значение → `422 invalid_request`). Позже её можно поменять через `PATCH /v1/campaigns/{id}`. Значение применяется как есть: потолка развёртывания, который урезал бы его, нет. При `auto_speed: true` не обязательна: если передана, сохраняется как ручная скорость, к которой кампания вернётся при выключении авто-скорости.
`auto_speed`	boolean	нет	`false`	Авто-скорость: скорость подбирает платформа, чтобы пройти базу до `ends_at`, пересчёт раз в 10 минут, не выше лимита сабгруппы, лимита одновременных звонков организации и 200 в минуту. Требует `ends_at` (`422 campaign_auto_speed_ends_at_required`); если окно кампании переходит через полночь в её зоне — ещё и `working_hours` (`422 campaign_auto_speed_working_hours_required`). Только настоящий boolean: `1` или `"true"` → `422 invalid_request`.

Поле	Тип	Обяз.	По умолч.	Описание
`planned_calls`	array	нет	`[]`	Номера для набора, см. ниже. Пусто — кампания создана без чего набирать (блокер `no_pending_contacts`, если только не задан `accepts_call_additions`).
`starts_at` / `ends_at`	number | null	нет	`null`	Окно кампании. `null` в начале означает «сейчас». `ends_at` должен быть позже `starts_at` (иначе `422 campaign_window_invalid`). **Дата** начала в прошлом (по календарной дате в зоне кампании) блокирует запуск (`starts_date_past`); прошедший `ends_at` — тоже (`ends_at_past`).
`timezone`	string	нет	`""`	Название зоны IANA (`Asia/Almaty`), в которой трактуются окно и рабочие часы. Не указана → зона организации. Неизвестная → `422 campaign_timezone_unknown`; фиксированные смещения не принимаются.
`working_hours`	array	нет	`[]`	Строки расписания `{"days": [0..6], "start": "ЧЧ:ММ", "end": "ЧЧ:ММ"}` (`days`: 0 — понедельник, непустой; `start` строго раньше `end`; строка никогда не переходит через полночь — ночная смена это две строки). Неверная форма → `422 campaign_spec_invalid`. Кампания может только **сужать** базовое окно вашей организации: под пакетом рынка США — законное окно юрисдикций контактов, иначе — окно обзвона организации; более широкая строка → `422 working_hours_wider_than_law`. **Пусто означает, что действует само базовое окно** — кампания ничего не сужает, а набор ограничен законным окном (предупреждение `working_hours_law_only`) либо окном обзвона организации (`working_hours_org_window_only`). Круглосуточного набора не бывает.
`call_strategy`	`"standard"` | `"expert"`	нет	`"standard"`	`standard` повторяет по фиксированной лестнице; `expert` решает по каждому исходу из правил `retry_config.rules`.
`retry_config`	object	нет	`{"attempts": 1, "interval_s": 900}`	См. «Настройку повторов» ниже в этом разделе. **По умолчанию — одна попытка, без перезвонов.**
`duplicate_action`	`"reject"` | `"keep_first"` | `"keep_last"`	нет	`"reject"`	Что делать, если один и тот же телефон встречается в `planned_calls` дважды (и внутри партии `calls`).

Поле	Тип	Обяз.	По умолч.	Описание
`classification`	`"marketing"` | `"servicing"` | `"debt_collection"`	нет	значение организации, иначе `"marketing"`	Чем звонит кампания. Определяет **тип согласия**, под которым ставится подпись (`marketing` → `pewc` , `servicing` и `debt_collection` → `pec`). Не указали — берётся классификация кампаний по умолчанию организации; если она не задана — `marketing` , самый строгий тип. Иное → `422 campaign_classification_invalid` .
`use_contact_timezone`	boolean	нет	`false`	Набирать каждый контакт в его собственной зоне (из колонки `timezone`), а не в зоне кампании.
`trunk_id` , `route_id` , `from_number`	string	нет	разрешается	Привязка набора, см. ниже.
`attestation`	object	нет	—	`{"confirmed": true}` — подтверждение согласия, необходимое до набора. При создании необязательно: без него кампания создаётся черновиком с блокером `attestation_required` .
`accepts_call_additions`	boolean	нет	`false`	Держать кампанию открытой к добавлению партий номеров (Управление кампанией): опустевшая очередь её не завершает, только `ends_at` или отмена. Требует `ends_at` (`422 campaign_open_ends_at_required`) не дальше горизонта оператора, по умолчанию 30 суток (`422 campaign_open_ends_at_too_far`). Задаётся только при создании.
`dnc`	`"block"` | `"skip"`	нет	`"block"`	Что делать с номерами из реестра «не звонить». Иное значение → `422 dnc_decision_invalid` .

Поля, которые у кампании есть, но через `/v1` **не задаются** (`frequency_override` , кабинетные пресеты), молча игнорируются.

Порядок отказов на создании

До того как кампания создана (слот идемпотентности освобождается, исправленный повтор с тем же ключом пройдёт):

1. `422 invalid_request` — тело не проходит схему (нет `name` , `agent_id` или `calls_per_minute` при выключенной авто-скорости, неверные типы).
2. `422 idempotency_key_required` , `422 dnc_decision_invalid` .
3. `422 idempotency_key_reuse` / `409 idempotency_key_in_flight` / переигровка прежнего ответа.
4. `422 campaign_window_invalid` , затем `422 campaign_auto_speed_ends_at_required` / `campaign_auto_speed_working_hours_required` , затем `422 campaign_open_ends_at_required` / `campaign_open_ends_at_too_far` .
5. `422 campaign_agent_unknown` , `422 campaign_timezone_unknown` .
6. Привязка набора: `422 dialing_route_unresolved` / `dialing_route_ambiguous` / `dialing_binding_incomplete` .
7. `422 working_hours_wider_than_law` .
8. Валидация хранилища: `422 campaign_spec_invalid` (скорость вне 1–200, настройка повторов, форма рабочих часов, действие по дублям, длина названия), `422 campaign_classification_invalid` .

После того как кампания создана (слот **остаётся занятым** — повтор с тем же ключом отвечает `409 idempotency_key_in_flight`; кампанию ищите через `GET /v1/campaigns`):

9. Запланированные звонки: `409 market_package_not_installed`, `502 contact_file_storage_unavailable`, `422 contact_rows_rejected` (строки не удалось сохранить вовсе — это не то же самое, что строки, отклонённые разборщиком: те приходят внутри `import` вместе с `201`).
10. Запуск: `409 campaign_transition_invalid` (неожиданная гонка состояний).

Успешный ответ — `201 Created`

Ответ **всегда** `201`, если кампания создана, независимо от того, начался ли набор. Единственный источник истины об этом — `status`.


```
{
  "campaign_id": "cam-b9b536a86e",
  "status": "running",
  "created_via": "api",
  "launched_via": "api",
  "name": "june collections",
  "agent_id": "collections-agent",
  "timezone": "Asia/Almaty",
  "timezone_source": "request",
  "starts_at": null,
  "ends_at": null,
  "accepts_call_additions": false,
  "calls_per_minute": 10,
  "call_strategy": "standard",
  "classification": "marketing",
  "consent_tier": "pewc",
  "retry_config": { "attempts": 1, "interval_s": 900 },
  "duplicate_action": "reject",
  "use_contact_timezone": false,
  "working_hours": [{ "days": [0, 1, 2, 3, 4], "start": "09:00", "end": "19:00" }],
  "binding": { "trunk_id": "trk-31d4dc745a", "route_id": "rt-out-a", "from_number": "77000000001" },
  "import": {
    "import_id": "imp-353d158dd2",
    "status": "accepted",
    "rows_seen": 2,
    "accepted": 2,
    "warned": 0,
    "rejected": 0,
    "duplicates_found": 0,
    "dnc_matched": 0,
    "refusal_code": "",
    "issues": {},
    "dnc_screen": { "status": "clean", "checked_rows": 2, "unscreened_rows": 0, "matched_rows": 0,
"suppressed_contacts": 0, "fingerprint": "20e2…", "items": [], "has_more": false, "hidden_rows": 0,
"all_dialable_suppressed": false, "as_of_launch": true, "screened_at": 1788941847.27, "audit_id": 2, "decision":
"clean", "import_id": "imp-353d158dd2" },
    "dnc_matches": []
  },
  "dnc_screen": { "status": "clean", "as_of_launch": true, "…": "тот же объект, что import.dnc_screen" },
  "blockers": [],
  "warnings": [
    { "code": "starts_immediately", "message": "…" },
    { "code": "frequency_policy_active", "rolling_7d_cap": 7, "daily_cap": 2, "source": "platform_default",
"message": "…" }
  ]
}
```

ТЕХНИЧЕСКИЙ СПРАВОЧНИК

Пример: перезвоны по исходам

Готовый запрос `POST /v1/campaigns` с экспертной логикой: свой интервал перезвона на «занято», «нет ответа», автоответчик и короткий разговор.

У каждого исхода звонка может быть своя пауза до следующего набора: правило экспертной логики с действием `retry_in` и параметром `params.interval_s`. Ниже — готовый запрос и что делает каждая его часть.

Запрос целиком

Кампания создаётся и запускается одним запросом `POST /v1/campaigns`.

```
BASE_URL="https://app.hubtalk.ai"   # хост, на котором работает ваш кабинет

curl -X POST "$BASE_URL/v1/campaigns" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -H "Content-Type: application/json" \
  -H "Idempotency-Key: campaign-2026-10-22-01" \
  -d '{
    "name": "Напоминание о платеже — октябрь",
    "agent_id": "payment_reminder",
    "calls_per_minute": 10,
    "timezone": "Asia/Almaty",
    "working_hours": [
      { "days": [0, 1, 2, 3, 4], "start": "09:00", "end": "19:00" }
    ],
    "classification": "debt_collection",
    "call_strategy": "expert",
    "retry_config": {
      "attempts": 6,
      "interval_s": 3600,
      "rules": [
        { "if": "busy",              "then": "retry_in",          "params": { "interval_s": 600,  "max_matches": 3
} },
        { "if": "no_answer",         "then": "retry_in",          "params": { "interval_s": 5400, "max_matches": 2
} },
        { "if": "voicemail_reached", "then": "retry_in",          "params": { "interval_s": 7200, "max_matches": 2
} },
        { "if": "short_call",        "then": "retry_in",          "params": { "max_duration_s": 15, "interval_s":
3600 } },
        { "if": "no_speech",         "then": "wait_for_pass_end", "params": { "max_matches": 1 } },
        { "if": "user_declined",     "then": "stop_calling" }
      ]
    },
    "attestation": { "confirmed": true },
    "planned_calls": [
      { "phone": "+77001234567", "name": "Айгерим", "variables": { "debt": "15000" } },
      { "phone": "+77007654321", "name": "Данияр",  "variables": { "debt": "8200" } }
    ]
  }'
```

В ответ приходит `201` с `campaign_id`, `status`, `blockers` и `warnings`. `status: "running"` — обзвон пошёл; `draft` — что-то мешает запуску, причина в `blockers`. Полный ответ — в Создании кампании.

Что делает каждое правило

Каждое правило — пара «если исход → то действие». Интервал считается от конца звонка и для этого контакта заменяет интервал кампании.

Исход (`if`)	Действие (`then`)	Через сколько	Сколько раз за контакт
`busy` — занято	`retry_in` — перезвонить через время	10 мин (`600`)	3
`no_answer` — не взяли трубку	`retry_in`	1 ч 30 мин (`5400`)	2
`voicemail_reached` — автоответчик	`retry_in`	2 ч (`7200`)	2
`short_call` — разговор короче 15 с (`max_duration_s`)	`retry_in`	1 ч (`3600`)	без предела правила, только `attempts` кампании
`no_speech` — соединилось, но абонент молчал	`wait_for_pass_end` — перезвонить на следующем проходе	—	1
`user_declined` — абонент отклонил звонок	`stop_calling` — больше не звонить	—	—

Правило на `voicemail_reached` срабатывает и на `voicemail_suppressed` — когда автоответчик определён, но сообщение оставлять было нельзя. Все значения `if` — это коды `status_reason` из Исходов звонка плюс `short_call` и `goal_not_achieved`. Исход, под который не подошло ни одно правило, решает общая таблица исходов, как в `standard`: автоответчик без своего правила перезванивается через `interval_s` кампании (Исходы звонка). Поэтому в примере на `voicemail_reached` стоит своё правило; когда его `max_matches` исчерпан, автоответчик дальше перезванивается по общей таблице, пока есть наборы. Чтобы на автоответчик больше не звонить, нужно `stop_calling`.

Правила и ограничения

- **`attempts` — общий потолок наборов на контакт, считая первый**, от 1 до 30. Правила его не расширяют: если наборы кончились, контакт закрывается, даже если у правила остались срабатывания.
- **`interval_s` у `retry_in`** — целые секунды от `60` до `86400` (от 1 минуты до 24 часов). У `retry_in` он обязателен, у других действий — запрещён, иначе `422 campaign_spec_invalid`.
- **`max_matches`** — сколько раз правило может сработать для одного контакта, от 1 до 30, по умолчанию 1. **У `short_call` его нет** — сервер откажет с `422 campaign_spec_invalid`.
- **`max_duration_s` у `short_call`** обязателен, не меньше `10` секунд.
- **Перезвон по постоянной причине запрещён.** Для `invalid_destination` и `spam_blocked` на всех рынках и для `user_declined` и `screener_declined` в пакете рынка США можно только `stop_calling`. На остальных рынках `user_declined` считается временным и перезвон на него разрешён — в примере мы сознательно ставим `stop_calling`.
- **Одна пара «условие + действие» — один раз.** Дубль отклоняется.
- **Экспертный режим без единого правила** не запустится: блокер `rules_empty`.
- **Окно обзвона, реестр «не звонить» и частотные лимиты** действуют на каждый перезвон. Перезвон, который не успевает до конца кампании (`ends_at`), не назначается.
- **Предупреждение `rule_interval_exceeds_end`** — интервал правила не меньше остатка окна кампании: такой перезвон не состоится. Запуск оно не блокирует.
- **`id` правила не передаём** — его назначает сервер (`r-xxxxxxxx`) и возвращает в ответе.

Чтобы обзвон начался сразу

- **`Idempotency-Key`** обязателен, для каждой новой кампании — свой. Повтор запроса с тем же ключом после таймаута не создаст вторую кампанию.
- **`attestation: {"confirmed": true}`** — подпись согласия абонентов. Без неё кампания создаётся черновиком с блокером `attestation_required`. Подробнее — в Согласии, реестре и ответственности.
- **Агент опубликован.** Иначе — черновик с блокером `agent_not_published`. `agent_id` — это **имя** агента.

- **У агента включено «Определять автоответчик».** Без него исхода `voicemail_reached` не будет, и правило на автоответчик не сработает ни разу.
- **`trunk_id`, `route_id` и `from_number` не передаём** — платформа сама найдёт линию по агенту (Привязка набора).
- **`working_hours` только сужает** окно обзвона организации; шире — `422 working_hours_wider_than_law`.

Где ещё посмотреть

Что	Страница
Заголовки, поля запроса, `call_strategy`, порядок отказов, пример ответа	Создание кампании
`retry_config` и правила: `if`, `then`, `params`	Поля кампании → Настройка повторов
Коды исходов для `if`	Исходы звонка
Блокеры и предупреждения запуска	Объект кампании
Экспертная логика в кабинете	Запуск кампании

ТЕХНИЧЕСКИЙ СПРАВОЧНИК

Объект кампании, блокеры и предупреждения

Что приходит в ответе о кампании и как читать блокеры и предупреждения запуска.

Объект кампании

Приходит в `POST /v1/campaigns`, `GET /v1/campaigns/{id}`, `POST …/launch`, `POST …/pause`, `POST …/resume`, `POST …/cancel`.

Поле	Тип	Описание
`campaign_id`	string	Идентификатор (`cam-…`).
`status`	string	`draft` | `scheduled` | `running` | `paused` | `completed` | `cancelled`.
`created_via`, `launched_via`	string	`api` | `web` | `""`. `launched_via` проставляется на переходе запуска и у черновика остаётся `""`.
`name`, `agent_id`	string	Как сохранено (название обрезано по пробелам).
`timezone`	string	**Действующая** зона IANA.
`timezone_source`	string	`request`, если зону назвали вы, и `organization`, если она унаследована. Одинаково на всех путях (создание, чтение, список).
`starts_at`, `ends_at`	number | null	Секунды эпохи.
`accepts_call_additions`	boolean	
`calls_per_minute`	integer	Скорость, с которой кампания набирает, звонков в минуту. Совпадает с тем, что вы прислали или задали позже. У кампании с авто-скоростью — ручная скорость, к которой она вернётся при выключении авто-скорости.
`auto_speed`	boolean	`true` — скорость подбирает платформа, см. авто-скорость.
`auto_rate_current`	integer | null	Темп, с которым идущая авто-кампания набирает после последнего пересчёта. `null` у ручной кампании и у кампании, которая не идёт.
`auto_rate_needed`	integer | null	Сколько звонков в минуту нужно, чтобы пройти базу до `ends_at`, — до всех лимитов. Больше `auto_rate_current` — лимиты не дают успеть. `null` там же, где и `auto_rate_current`.
`call_strategy`, `classification`	string	
`consent_tier`	string	`pewc` | `pec` — выводится из `classification`, не хранится; тип, под формулировкой которого стоит ваша подпись.
`retry_config`	object	Нормализованная форма; идентификаторы правил присвоены.
`duplicate_action`, `use_contact_timezone`, `working_hours`		Нормализованные отражения присланных настроек.
`binding`	object	`{"trunk_id", "route_id", "from_number"}` — действующая привязка; `route_id` равен `""`, когда группа разрешается из транка при запуске, а не хранится.
`import`	object | null	Сводка загрузки `planned_calls` **этого запроса** (только при создании). `null`, если `planned_calls` был пуст, и на всех путях чтения — сводка живёт в `GET /v1/imports/{import_id}`.
`dnc_screen`	object | null	Сверка текущей (запусковой) загрузки кампании: живая у черновика, зафиксированная после запуска. `null`, если у кампании нет загрузки.
`blockers`, `warnings`	array	Есть при создании, чтении и `launch`; **отсутствуют** у `pause` / `resume` / `cancel`.

Как читать ответ на создание

1. `status` — `running` (набор идёт), `scheduled` (окно в будущем) или `draft` (не начался). Ничто другое в теле не означает «запустилось».
2. Непустой `blockers` **при создании** ⟹ `status: "draft"`. Кампания и отчёт по загрузке **сохраняются**. Исправьте то, что названо в блокерах, и вызовите `POST /v1/campaigns/{id}/launch`.
3. `import` равен `null`, если `planned_calls` был пуст; в `blockers` тогда лежит `no_pending_contacts` (если кампания не открыта к добавлению звонков).
4. `warnings` ничего не блокируют. Два из них есть практически у каждой кампании: `starts_immediately` (нет будущего `starts_at`) и `frequency_policy_active` (действующие ограничения частоты на номер) — последний заменяется на `frequency_limits_disabled`, когда эти ограничения выключены.
5. `timezone_source` — `request`, если зону назвали вы, и `organization`, если она унаследована. Читается одинаково на всех путях: ответ на создание, `GET /v1/campaigns/{id}` и список согласованы.

Блокеры и предупреждения

`blockers` и `warnings` считает та же проверка запуска, что работает в кабинете. Каждый элемент — объект с машиночитаемым `code`; у большинства блокеров есть ещё `field` — поле запроса, о котором идёт речь.

Форма зависит от пути — читайте только то, что гарантировано:

Где	Форма элемента	Примечания
успешный объект кампании (`POST /v1/campaigns`, `GET /v1/campaigns/{id}`, `POST …/launch`, `GET …/launch-check`)	`{"code", "field"?, "message", …доп. поля}`	`message` — **русское человекоориентированное пояснение, в контракт не входит**: не разбирайте его и не показывайте конечным пользователям. Дополнительные поля контекста по каждому коду перечислены ниже и стабильны.
`409 launch_blocked` (`error.blockers`)	`{"code", "field", "dnc_screen"?}`	Только коды и поля; блокеры по реестру дополнительно несут обновлённую `dnc_screen`.
`201` у `POST …/calls`	`{"code", "field"}`	Только коды и поля (`field` может быть `""`).

> **Как читать `blockers` в `GET /v1/campaigns/{id}` и `launch-check`.** На путях чтения список отвечает на вопрос «что сказала бы проверка запуска прямо сейчас», причём считается **без** подписи и независимо от текущего статуса. Следствия: у работающей кампании всегда виден `attestation_required` (подпись даётся на каждый запуск, а не хранится как состояние), а у работающей кампании с опустевшей очередью — `no_pending_contacts`. То есть на путях чтения непустой `blockers` **не** означает `status: "draft"` — для этого есть `status`, а `blockers` читайте как список того, что должен будет пройти будущий `launch` / `resume`.

Коды блокеров

Код	`field`	Доп. поля	Что означает
`no_pending_contacts`	`contacts`		Нет контактов, ждущих набора. Не поднимается у кампании с `accepts_call_additions: true`.
`agent_not_published`	`agent_id`		У агента нет опубликованной версии.
`binding_invalid`	`trunk_id`		Транк, группа или номер привязки недоступны (агента нет в группе, пуст пул номеров, нет гранта на транк…).
`starts_date_past`	`starts_at`		**Дата** начала (в зоне кампании) в прошлом. Прошедшее *время* сегодняшней даты — это только предупреждение `starts_immediately`.
`ends_before_starts`	`ends_at`		`ends_at` ≤ `starts_at`.

Код	field	Доп. поля	Что означает
`ends_at_past`	`ends_at`		`ends_at` уже прошёл.
`policy_unavailable`	`market_profile`		Политика набора (пакет рынка) вашей организации не установлена; обратитесь в поддержку.
`contacts_missing_geo`	`contacts`		Рынок США: у контактов загрузки нет геолокации.
`contacts_geo_rejected`	`contacts`	`rejected`, `filename`	Рынок США: строки текущей загрузки отклонены из-за непригодной локации. Снимается только новой загрузкой.
`dnc_screen_required`	`contacts`	`dnc_screen`	У текущей загрузки нет записанной построчной проекции реестра: загрузите контакты заново.
`dnc_matches`	`contacts`	`dnc_screen`, `registry_changed`	Совпадения с реестром и нет действующего решения `skip` для этого точного состава. `registry_changed: true` означает, что решение было, но реестр с тех пор изменился.
`all_contacts_suppressed`	`contacts`	`dnc_screen`	Все ожидающие контакты в реестре; пропустить нельзя.
`attestation_required`	`attestation`		В этом запросе нет подтверждения согласия.
`attestation_stale`	`attestation`		`attestation.import_id` — не текущая загрузка кампании.
`rules_empty`	`retry_config`		Стратегия `expert` без правил.
`rule_permanent_retry`	`retry_config`	`rule_id`	Правило повторяет постоянный исход (`invalid_destination`, `spam_blocked`; `user_declined` и `screener_declined` под профилем США).
`signature_columns_missing`	`contacts`	`agent`, `variables`	В контактах нет колонки для одной или нескольких сигнатурных переменных агента (отмеченных у агента **«В сигнатуре»**); `variables` их перечисляет. Сверка идёт с текущей опубликованной версией агента; у `POST …/calls` — с колонками самой партии. Переменные вне сигнатуры не проверяются.
`campaign_paused`	`status`		**Только у** `POST …/calls`: партия принята, но до `resume` ничего не набирается.
`auto_speed_ends_at_required`	`ends_at`		Авто-скорость без `ends_at`.
`auto_speed_working_hours_required`	`working_hours`		Авто-скорость у кампании на несколько дней без рабочих часов.

Коды предупреждений

Код	Доп. поля	Что означает
`starts_immediately`		Нет будущего `starts_at`: набор начнётся при запуске.
`frequency_policy_active`	`rolling_7d_cap`, `daily_cap`, `source` (`organization` | `platform_default`)	Действующие для этой кампании ограничения частоты на номер. Присутствует, пока лимиты включены, — то есть у каждой кампании, кроме случая ниже.

Код	Доп. поля	Что означает
`frequency_limits_disabled`	`source` (`campaign` | `organization`)	Настраиваемые частотные лимиты выключены, `source` говорит какой настройкой. Приходит **вместо** `frequency_policy_active`, не вместе с ним. На американском профиле рынка Reg F и капы штатов продолжают действовать; на казахстанском профиле законного слоя нет.
`working_hours_law_only`	`field: "working_hours"`	Рынок США, пустое расписание: действует только законное окно.
`working_hours_org_window_only`	`field: "working_hours"`, `start`, `end`, `source`	Пустое расписание: действует окно обзвона организации `start` – `end`.
`retry_schedule_exceeds_end`	`lost`	Лестница повторов длиннее остатка окна кампании; `lost` повторов не состоится.
`variable_case_mismatch`	`variable`, `column`	Колонка сопоставлена с сигнатурной переменной без учёта регистра и разделителей.
`variable_partial`	`variable`	Сигнатурная переменная заполнена только у части строк.
`rule_action_unsupported`	`rule_id`	Правило `expert` использует действие, которое движок не исполняет; решает общая таблица исходов. Все три действия — `wait_for_pass_end`, `stop_calling` и `retry_in` — исполняются, поэтому сейчас это предупреждение не выдаётся.
`rule_goal_without_criteria`	`rule_id`	Правило `goal_not_achieved` при том, что агент не задаёт критерия успеха — правило никогда не сработает.
`rule_budget_masked`	`rule_id`	`max_matches` правила не меньше числа попыток кампании — недостижимо.
`rule_interval_exceeds_end`	`rule_id`, `interval_s`, `window_s`	Интервал правила `retry_in` не короче остатка окна кампании — этот повтор не состоится.
`auto_speed_exceeds_subgroup_limit`	`field: "trunk_id"`, `needed`, `limit`	Авто-скорость: чтобы успеть до `ends_at`, нужно `needed` звонков в минуту, а сабгруппа допускает `limit` — часть базы не будет дозвонена.
`auto_speed_exceeds_concurrency_limit`	`field: "ends_at"`, `needed`, `limit`, `slots`	То же, но ограничивает лимит одновременных звонков организации: `slots` звонков дают около `limit` в минуту.
`auto_speed_exceeds_platform_limit`	`field: "ends_at"`, `needed`, `limit`	То же, но ограничивает потолок платформы — 200 в минуту.
`manual_speed_exceeds_subgroup_limit`	`field: "trunk_id"`, `speed`, `limit`	Ручная скорость выше лимита сабгруппы: наборы сверх него будут отклоняться и повторяться.
`manual_speed_exceeds_concurrency_limit`	`field: "limits"`, `speed`, `limit`, `slots`	Ручной скорости нужно больше одновременных звонков, чем есть у организации: `slots` дают около `limit` в минуту.

ТЕХНИЧЕСКИЙ СПРАВОЧНИК

Поля кампании

Повторы, номера, привязка набора, подтверждение согласия, сверка с реестром и идемпотентность.

Настройка повторов

`retry_config` нормализуется при сохранении и возвращается нормализованным.

Поле	Тип	По умолч.	Правила
`attempts`	integer	`1`	Всего попыток набора на контакт, 1–30. `0` читается как `1`; больше 30 — ограничивается до 30; отрицательное → `422 campaign_spec_invalid`. **При значении по умолчанию ни один номер не перезванивается.** У мастера кампании в кабинете тот же предел — до 29 повторов (30 наборов) — и готовые интервалы от 1 минуты до 7 дней; API принимает любой положительный `interval_s`.
`interval_s`	integer	`900`	Пауза между попытками в секундах. Должна быть `> 0`, когда `attempts > 1`, иначе `422 campaign_spec_invalid`.
`retry_mode`	`"regular"` | `"custom"`	`regular`	`custom` использует лестницу `intervals_s` по каждому повтору.
`intervals_s`	array of integers	—	Секунды до каждого повтора в режиме `custom` (подгоняется под `attempts − 1`).
`rules`	array	—	Только `expert`. Правило: `{"id"?, "if", "then", "params"?}`. `if` — значение `status_reason` (Исходы звонка) либо `short_call` / `goal_not_achieved`; `then` — `wait_for_pass_end`, `stop_calling` или `retry_in`; `params`: `max_matches` (1–30, по умолчанию 1) для любого условия, кроме `short_call` — там он отклоняется с `422 campaign_spec_invalid`, а короткие звонки ограничивает только `attempts` кампании, — `max_duration_s` (≥ 10, обязателен) для `short_call`, `interval_s` для `retry_in` — целые секунды от 60 до 86 400 (от минуты до 24 часов), обязателен у этого действия и запрещён у остальных, иначе `422 campaign_spec_invalid`. По `retry_in` следующая попытка — не раньше `interval_s` от окончания звонка: интервал правила заменяет для этого контакта интервал кампании, окно обзвона, реестр «Не звонить» и лимиты проверяются как у любого повтора. Правило `retry_in` без `interval_s`, сохранённое раньше, берёт интервал кампании. Идентификаторы присваивает сервер (`r-xxxxxxxx`); присланные вами, если это не отголоски существующих, заменяются. Голый массив правил принимается как устаревшая форма.

Свой интервал перезвона для каждого исхода

При `call_strategy: "expert"` у каждого исхода может быть своя пауза до следующего набора: правило `retry_in` с `params.interval_s`. Исход, под который не подошло ни одно правило, решает общая таблица исходов, как в `standard` (Исходы звонка): например, автоответчик без своего правила перезванивается через `interval_s` кампании.

```
{
  "call_strategy": "expert",
  "retry_config": {
    "attempts": 6,
    "interval_s": 3600,
    "rules": [
      { "if": "busy", "then": "retry_in", "params": { "interval_s": 600, "max_matches": 3 } },
      { "if": "no_answer", "then": "retry_in", "params": { "interval_s": 5400, "max_matches": 2 } },
      { "if": "voicemail_reached", "then": "retry_in", "params": { "interval_s": 7200, "max_matches": 2 } },
      { "if": "short_call", "then": "retry_in", "params": { "max_duration_s": 15, "interval_s": 3600 } },
      { "if": "no_speech", "then": "wait_for_pass_end", "params": { "max_matches": 1 } },
      { "if": "user_declined", "then": "stop_calling" }
    ]
```

```
  }
}
```

Правило	Что происходит
`busy`	Перезвонить через 10 минут, не больше 3 раз.
`no_answer`	Перезвонить через 1 час 30 минут, не больше 2 раз.
`voicemail_reached`	Перезвонить через 2 часа, не больше 2 раз.
`short_call`	Разговор короче 15 секунд — перезвонить через час.
`no_speech`	Абонент молчал — перезвонить один раз, на следующем проходе по списку.
`user_declined`	Больше не звонить на этот номер.

- `attempts` ограничивает наборы на контакт по всем правилам вместе: при `6` номер набирается не больше шести раз, сколько бы ни давали бюджеты правил в сумме.
- Перезвон, который пришёлся бы после `ends_at` кампании, не назначается; такие правила называет предупреждение `rule_interval_exceeds_end` (Объект кампании).
- Действие-перезвон на постоянную причину (`invalid_destination` , `spam_blocked`) отклоняется.
- `expert` с пустым массивом `rules` не запустится: будет блокер `rules_empty` .

Полный запрос с этой настройкой — в Примере: перезвоны по исходам.

Номера: `planned_calls`

Каждый объект — одна строка списка контактов, и **его ключи и есть имена колонок**. Понимается всё, что понимает загрузка файла: колонки пакета рынка (`contact_zip` , `contact_state` , `contact_city`), колонки договорных групп (`contract_id` , `role`), `timezone` , `name` и любая переменная, которую читает сценарий агента. Вложенный объект `variables` — синтаксический сахар, он раскладывается в те же колонки (ключ верхнего уровня побеждает такой же ключ внутри `variables`). Значения приводятся к тексту (логические → `true` / `false` , объекты → компактный JSON), поэтому ведущие нули сохраняются. Голые `zip` , `state` и `city` — обычные переменные, а не локация.

```
{
  "planned_calls": [
    {"phone": "+77001234567", "name": "Aigerim", "variables": {"debt": "15000"}},
    {"phone": "+77007654321", "name": "Daniyar", "contact_zip": "73301"}
  ]
}
```

Колонка `phone` обязательна в каждой строке; значение меньше чем с десятью цифрами отклоняется как `phone_unreadable` — отклоняется **строка**, а не запрос. Построчные коды замечаний — в Отчёт по загрузке номеров.

Привязка набора

Внутренних идентификаторов транков и групп у вас нет, поэтому платформа разрешает их сама: **не присылайте `trunk_id` , `route_id` и `from_number` вместе** — и группы, привязанные к вашему агенту, будут найдены по всем транкам, видимым вашей организации.

Итог	Результат
ровно одна группа	используется она; номер на каждой попытке выбирается по политике исходящего номера группы
ни одной	`422 dialing_route_unresolved`
больше одной	`422 dialing_route_ambiguous` , кандидаты названы в сообщении парами `trunk_id/route_id`

Неоднозначность никогда не решается выбором первого кандидата: набор с другого транка потратил бы другой номер и другую квоту, а узнали бы вы об этом по счёту. Прочитать кандидатов:

```
GET /v1/dialing-routes?agent_id=collections-agent
```

Ответ — `{"dialing_routes": [{"trunk_id", "trunk_name", "route_id", "route_name", "agents", "caller_numbers"}]}` — без настроек подключения транка и без учётных данных. Пустой `agent_id` перечисляет все маршруты, видимые вашей организации.

> **`trunk_id` — якорь.** Назовите его, и остальное следует существующим правилам платформы: группа разрешается из транка и агента при запуске, а номер выбирает сама группа по своей политике исходящего номера. `route_id` или `from_number` **без** `trunk_id` отклоняются кодом `422 dialing_binding_incomplete`: платформа не станет угадывать, какой транк вы имели в виду. Транк, названный явно, но непригодный (агента нет в его группе, пуст пул), при создании не отклоняется — он становится блокером `binding_invalid`.

Подтверждение согласия

Для набора нужно подтверждение согласия: `{"attestation": {"confirmed": true}}`. Запись хранится неизменяемо против той версии данных, которую покрывает, и против формулировки типа согласия кампании, а подписантом выступает ваш API-ключ — выпуск ключа и есть человеческое действие, которому приписывается подпись (платформа записывает, кто и когда выпустил ключ).

Подтверждение одной загрузки **не** переносится на новую; поэтому в `POST …/launch` поле `attestation.import_id` обязательно (→ `422 attestation_import_id_required`), а `import_id`, не являющийся текущей загрузкой кампании, даёт блокер `attestation_stale`. Партии добавляемых звонков подписываются отдельно.

Сверка с реестром «не звонить»: `dnc_screen`

`dnc_screen` — авторитетный результат. Проверяется каждая читаемая исходная строка до отбраковки по рынку и до дедупликации, поэтому `matched_rows` может быть больше `suppressed_contacts`.

Поле	Тип	Что означает
`status`	string	`clean` | `matches` | `screen_unavailable` (у загрузки нет записанной построчной сверки).
`checked_rows`	integer | null	Читаемые строки, которые проверены.
`unscreened_rows`	integer | null	Строки с нечитаемым телефоном, не проверены.
`matched_rows`	integer | null	Строки, чей номер есть в реестре (до дедупликации).
`suppressed_contacts`	integer | null	Уникальные совпавшие номера.
`fingerprint`	string	Детерминированный хеш точного состава совпадений; решение `skip` привязано к нему.
`items`	array	До 200 совпадений: `{"row_number", "phone", "phone_key", "contact_name", "dnc_entry_id", "source", "reason", "recorded_at"}`.
`has_more`, `hidden_rows`		Обрезан ли `items` и насколько.
`all_dialable_suppressed`	boolean	Все ожидающие контакты совпали.
`as_of_launch`	boolean	`false` — живая сверка с реестром; `true` — зафиксированный при запуске снимок.
`screened_at`, `audit_id`	number | null, integer | null	Когда снят зафиксированный снимок и его аудит-запись.

Поле	Тип	Что означает
`decision`, `import_id`	string	Только у зафиксированного снимка: `clean` | `skip` и загрузка, которую он покрывает.

`import.dnc_matches` — совместимая проекция `items` (те же объекты плюс алиас `row` для `row_number`); `import.dnc_matched` равно `matched_rows` (`0`, когда сверка недоступна).

Совпадения останавливают запуск блокером `dnc_matches`. Присланное `dnc: "skip"` записывает то же аудируемое решение, что человек принимает в кабинете. Оно действует для этой конкретной загрузки **и этого** `fingerprint`: новое совпадение или заменённая запись реестра делают решение устаревшим, и `launch-check` / `409` возвращают пересчитанную сверку. Если совпали все ожидающие контакты, `all_contacts_suppressed` остаётся безусловным. `skip` при чистой сверке ничего не записывает, и запуск идёт как обычно.

Значение по умолчанию `dnc: "block"` при явном запуске тоже записывается решением, и более поздний `block` отзывает любой прежний `skip`. Поэтому явный запуск без поля `dnc` никогда не едет на пропуске, который кто-то одобрил в кабинете: присылайте `dnc: "skip"` сами, если запуск означает именно это. Решению нужна загрузка, к которой применяться: `dnc` при запуске кампании без загрузки → `409 dnc_decision_without_upload`; на загрузке без записанной проекции → `409 dnc_screen_required`.

До первого запуска сверка и отчёт только по совпадениям следуют за живым реестром. Операция запуска пересчитывает сверку непосредственно перед переходом и атомарно фиксирует все совпавшие строки вместе с отметкой загрузки, подписью и состоянием кампании. После запуска чтения и отчёты используют этот снимок (`as_of_launch: true`), даже когда реестр меняется. Контакты, совпавшие по ходу кампании, всё равно снимаются в момент набора; `progress` считает их отдельно.

Идемпотентность запросов по кампаниям

`POST /v1/campaigns` и `POST /v1/campaigns/{id}/calls` работают по одному протоколу:

- Ключ занимается **до** начала работы, атомарно. Второй запрос с тем же ключом, пока первый ещё выполняется, получает `409 idempotency_key_in_flight` — дождитесь первого ответа, а не повторяйте снова.
- Тело фингерпринтится вместе с ключом (канонический JSON, порядок ключей не важен). Тот же ключ с **другим** телом — `422 idempotency_key_reuse`, и никогда не молчаливая переигровка прежнего ответа.
- Тот же ключ с тем же телом переигрывает сохранённый `201` байт в байт.
- Ключи ограничены вашим API-ключом, эндпоинтом и — для `calls` — кампанией, и помнятся сутки.
- При создании отказ **до** появления кампании освобождает ключ; отказ **после** — оставляет занятым. У `calls` любой отказ освобождает ключ.

ТЕХНИЧЕСКИЙ СПРАВОЧНИК

Управление кампанией

Запуск, пауза, отмена, прогресс, статистика и добавление партий номеров.

Метод и путь	Что делает
`GET /v1/campaigns`	список, новые первыми: `limit` (1–200, по умолчанию 50) и `cursor` → `{campaigns, has_more, next_cursor}`. Элементы — краткие представления: `campaign_id, status, name, agent_id, created_via, launched_via, starts_at, ends_at, timezone, timezone_source, accepts_call_additions, auto_speed, auto_rate_current, auto_rate_needed` — без блокеров
`GET /v1/campaigns/{id}`	объект кампании (`import: null`) с живой/зафиксированной `dnc_screen` и `blockers` / `warnings` пути чтения
`GET /v1/campaigns/{id}/launch-check`	`{blockers, warnings, dnc_screen}` — `dnc_screen` здесь всегда **живая**
`POST /v1/campaigns/{id}/launch`	`{"attestation": {"confirmed": true, "import_id": "imp-…"}, "dnc": "block"}`. Успех — `200` с объектом кампании (`blockers: []`); отказ — `409 launch_blocked`
`POST /v1/campaigns/{id}/calls`	добавить партию номеров, см. ниже
`POST /v1/campaigns/{id}/pause`	`running` → `paused`. Уже набранные звонки доигрывают, новых наборов нет
`POST /v1/campaigns/{id}/resume`	`paused` → `running`. Сначала заново прогоняет проверки запуска — за время паузы могло измениться что угодно
`POST /v1/campaigns/{id}/cancel`	любой нефинальный статус → `cancelled`. Необратимо; уже набранные звонки доигрывают
`PATCH /v1/campaigns/{id}`	сменить скорость обзвона или режим авто-скорости: `{"calls_per_minute": N}`, `{"auto_speed": true}` или оба, см. ниже
`GET /v1/campaigns/{id}/progress`	состояния контактов, удержания, темп, счётчики реестра и частоты
`GET /v1/campaigns/{id}/rate-history`	журнал пересчётов авто-скорости, см. ниже

Статусы. `draft` → (запуск) → `scheduled` (когда `starts_at` в будущем) или `running`; `scheduled` становится `running`, когда открывается окно; `running` ⇄ `paused`; `running` → `completed`, когда очередь опустела (ничего не ожидает и ничего не в наборе) — **кроме** кампании с `accepts_call_additions: true`, которая завершается только по наступлении `ends_at`; любой нефинальный статус → `cancelled`. `completed` и `cancelled` финальны. Наступление `ends_at` закрывает любую кампанию.

На `resume` блокеры `no_pending_contacts`, `attestation_required`, `attestation_stale`, `contacts_geo_rejected` и `dnc_screen_required` не применяются.

Ответы `pause` / `resume` / `cancel` содержат объект кампании **без** `blockers` и `warnings`: эти пути не запускают проверки запуска, а пустой массив читался бы как «проверено, всё чисто». Нужны проверки — вызовите `launch-check`.

Повторить переход — не ошибка. Кампания, уже находящаяся в целевом состоянии, отвечает `200` со своим текущим объектом: повтор после сетевого таймаута — не второе намерение. Переход, невозможный **из** текущего состояния (возобновить отменённую, поставить на паузу черновик), — `409 campaign_transition_invalid`. Разница между «уже сделано» и «сделать нельзя» — ровно та, которая вам нужна.

Заблокированный запуск или возобновление — ошибка, а не `200`. Создание отвечает `201`, потому что кампания и отчёт действительно созданы; `launch` — равно как и `resume`, который прогоняет те же проверки, — ничего не запустивший, отвечает `409 launch_blocked`, и причины едут в том же ответе:

```
{
  "error": {
```

```
    "type": "conflict_error",
    "code": "launch_blocked",
    "status": 409,
    "message": "The campaign did not start: error.blockers names every check that did not pass",
    "blockers": [
      {"code": "agent_not_published", "field": "agent_id"},
      {"code": "dnc_matches", "field": "contacts", "dnc_screen": {"status": "matches", "…": "…"}}
    ]
  }
}
```

Правило по всей поверхности кампаний: **действие, которое ничего не изменило, — всегда ошибка; успешный ответ означает, что состояние в теле и есть новое состояние.**

Смена скорости обзвона

`PATCH /v1/campaigns/{id}` с телом `{"calls_per_minute": N}` меняет скорость кампании в статусах `draft`, `scheduled`, `running` и `paused`. `N` — любое целое от 1 до 200, как при создании. Второе поле тела — `auto_speed`: `true` включает авто-скорость, `false` возвращает кампанию к `calls_per_minute`. Можно прислать одно поле или оба, хотя бы одно обязательно. Это единственные настройки уже созданной кампании, которые можно поменять через `/v1`.

```
curl -s -X PATCH "$BASE_URL/v1/campaigns/cam-7f3a"   -H "Authorization: Bearer $CFLOW_API_KEY"   -H "Content-Type: application/json"   -d '{"calls_per_minute": 30}'
```

- **Когда действует.** Идущая кампания (`running`) набирает с новой скоростью со следующего цикла набора, без перезапуска; на паузе (`paused`) — после возобновления; запланированная (`scheduled`) — с момента старта. У черновика значение просто сохраняется.
- **Ответ.** `200` с объектом кампании, как у `pause` / `resume`: без `blockers` и `warnings`.
- **Повтор того же значения** отвечает `200` и ничего не меняет. Запрос идемпотентен, `Idempotency-Key` не нужен.
- **Аудит.** Каждое реальное изменение записывается с прежним и новым значением и статусом кампании, от имени вашего ключа.
- **Кампании адаптера совместимости.** Адаптер задаёт скорость один раз — из первого запроса дня. Изменение отсюда действует до закрытия кампании этого дня; кампания следующего дня снова берёт скорость из запроса адаптера.

Статус	`error.code`	Причина
404	`campaign_not_found`	Такой кампании нет в вашей организации.
409	`campaign_closed`	Кампания `completed` или `cancelled` — в том числе если закрылась, пока шёл запрос.
422	`invalid_request`	Любое поле кроме `calls_per_minute` и `auto_speed`, пустое тело, нецелая скорость или `auto_speed` не boolean.
422	`campaign_spec_invalid`	Целое вне диапазона 1–200.
422	`campaign_auto_speed_ends_at_required`	`auto_speed: true` у кампании без `ends_at`.
422	`campaign_auto_speed_working_hours_required`	`auto_speed: true` у кампании на несколько дней без `working_hours`.

Скорость можно поменять и в кабинете — см. Скорость обзвона после запуска.

Добавление номеров в существующую кампанию

```
POST /v1/campaigns/{id}/calls
Idempotency-Key: <уникальная строка>        (обязателен)

{
```

```
  "calls": [
    {"phone": "+77001234567", "timezone": "Asia/Almaty", "variables": {"debt": "15000"}},
    {"phone": "+77007654321", "name": "Daniyar"}
  ],
  "attestation": {"confirmed": true}
}
```

Кампании принадлежат агент, привязка набора, правила повторов и рабочие часы; партия несёт только **номера, переменные и подпись**. Каждый элемент устроен как строка `planned_calls` (Поля кампании) — его ключи и есть имена колонок. Неизвестные поля верхнего уровня игнорируются: `agent_id` в теле не меняет агента кампании, а ключ `dnc` не обходит реестр.

Партия принимается в любую кампанию, которая не `completed` и не `cancelled`, — то есть в `draft`, `scheduled`, `running` и `paused`, независимо от того, открыта ли она к добавлению звонков. Закрытая отвечает `409 campaign_closed`.

Порядок отказов, все до разбора хотя бы одной строки: `422 idempotency_key_required` → `422 calls_required` (пустой `calls`) → `422 calls_batch_too_large` → `422 attestation_invalid` → `404 campaign_not_found` → `409 campaign_closed` → идемпотентность (`422 idempotency_key_reuse` / `409 idempotency_key_in_flight` / переигровка). Затем разбираются строки: `409 market_package_not_installed`, `422 contact_rows_rejected`, `502 contact_file_storage_unavailable` — как при создании, — и два отказа партии ниже.

> **Партия принимается целиком или отклоняется целиком.** Одна строка с ошибкой данных отклоняет всю партию кодом `422 calls_rejected`; один номер из реестра «не звонить» — кодом `422 calls_dnc_matched`, отдельным, чтобы ваша система отличала «исправить строки» от «убрать этих людей из списка». Если в партии проблемы обоих видов, побеждает `calls_rejected`. Структурные отказы разборщика (нет колонки телефона, нет пригодных строк, …) — тоже `calls_rejected`. Для отклонённой партии контакты не создаются и подпись не записывается.

Оба отказа несут `error.import`:

```
{
  "error": {
    "type": "invalid_request_error",
    "code": "calls_rejected",
    "status": 422,
    "message": "…",
    "import": {
      "import_id": "imp-b87d984fe7",
      "status": "refused",
      "refusal_code": "calls_rejected",
      "rows_seen": 2,
      "accepted": 0,
      "warned": 0,
      "rejected": 1,
      "dnc_matched": 0,
      "issues_total": 1,
      "issues": [{"row_number": 2, "code": "phone_unreadable", "severity": "error"}]
    }
  }
}
```

Здесь `issues` несёт первые 50 замечаний; полный отчёт по строкам — `GET /v1/imports/{import_id}/report` (и `?only=dnc_match` для совпадений с реестром). Отклонённая партия сохраняется записью загрузки со `status: "refused"` ровно для этого.

Успех — `201`:

```
{
  "campaign": {
    "campaign_id": "cam-…", "status": "running", "name": "june collections",
    "agent_id": "collections-agent", "created_via": "api", "launched_via": "api",
    "starts_at": null, "ends_at": null, "timezone": "Asia/Almaty",
    "timezone_source": "request", "accepts_call_additions": true
  },
  "import": { "import_id": "imp-…", "status": "accepted", "rows_seen": 2, "accepted": 2, "warned": 0, "rejected":
0, "duplicates_found": 0, "dnc_matched": 0, "refusal_code": "", "issues": {}, "dnc_screen": {"status": "clean",
"as_of_launch": false, "…": "…"}, "dnc_matches": [] },
  "blockers": [{"code": "campaign_paused", "field": "status"}],
  "warnings": [{"code": "starts_immediately", "field": ""}, {"code": "frequency_policy_active", "field": ""}]
}
```


`campaign` — краткое представление (тот же объект, что элемент списка). `blockers` означает «принято, но сейчас не набирается, и вот почему»: те же проверки запуска, что у `launch-check`, минус те, которые свежая партия делает неактуальными (`no_pending_contacts`, `attestation_required`, `dnc_screen_required`, `all_contacts_suppressed`, `dnc_matches`), плюс `campaign_paused`. Элементы на этом пути несут только `code` и `field`.

Правила, которые стоит знать, прежде чем гнать номера из CRM:

- **Предел** — до 1000 строк за запрос по умолчанию (настройка развёртывания): больше — `422 calls_batch_too_large`, пустая партия — `422 calls_required`. Один запрос считается один раз в лимите ключа.
- **Подпись** — `attestation` должна быть ровно `{"confirmed": true}`: без других ключей и без `import_id` (иначе `422 attestation_invalid`). Каждая принятая партия подписывается отдельно и привязана к своей загрузке; подпись запуска не переиспользуется для строк, которых на момент запуска не существовало.
- **Часовой пояс** — в кампании с `use_contact_timezone: true` под казахстанским рыночным профилем строка без `timezone` — это ошибка данных (`timezone_required`), и она отклоняет партию: платформа не подставит зону кампании строке, на которую никто не смотрел. У загрузки файла и у `planned_calls` при создании подстановка сохраняется.
- **Дубли** — внутри партии `duplicate_action` работает так же, как для файла. Номер, у которого в кампании уже есть контакт — даже уже набранный, — **принимается как новый контакт и набирается снова**: для адаптера, который сам управляет перезвонами, повтор *и есть* перезвон. Если перезвонами управляет ваша система, оставьте `retry_config.attempts = 1`; при большем бюджете перезвоны станут двухуровневыми.
- **Партия никогда не становится запусковой загрузкой кампании** — зафиксированный снимок реестра, подпись запуска и `dnc_screen` в `GET /v1/campaigns/{id}` продолжают указывать на загрузку, с которой кампанию запускали. Своя сверка партии — в теле `201` и в `GET /v1/imports/{import_id}`.
- **Идемпотентность** — ключ ограничен рамками кампании, поэтому один и тот же ключ с одним и тем же телом, отправленный в две кампании, — это две партии. Отклонённая партия **освобождает** ключ: исправьте строки и пришлите заново с тем же ключом (создастся вторая отклонённая запись загрузки — диагностически полезно, звонков это не касается). Повтор принятой партии переигрывает ответ; тот же ключ с другим телом — `422 idempotency_key_reuse`.

Кампании, открытые к добавлению номеров

Обычная кампания завершает себя сама, едва очередь опустела, — а завершённая кампания звонков не принимает. Чтобы кампания ждала номера, создайте её с `"accepts_call_additions": true`. Такая кампания:

- **требует** `ends_at` (`422 campaign_open_ends_at_required`) не дальше горизонта оператора — по умолчанию 30 суток (`422 campaign_open_ends_at_too_far`);
- может быть создана вообще без `planned_calls` и стартует пустой (блокера `no_pending_contacts` нет);
- закрывается только наступлением `ends_at` или отменой, но не опустевшей очередью.

Флаг задаётся при создании и потом не меняется; он возвращается в объекте кампании и в списке.

Прогресс и состояния контактов

`GET /v1/campaigns/{id}/progress`:

```
{
  "status": "running",
  "total": 120,
  "by_state": {"pending": 80, "in_progress": 3, "done": 30, "exhausted": 5, "suppressed": 2},
  "ready": 78,
  "holds": 2,
  "held_reasons": {"quiet_hours": 2},
  "suppressed_reasons": {"dnc_suppressed": 2},
  "dnc_suppressed_at_launch": 1,
  "dnc_suppressed_mid_campaign": 1,
  "backpressure": 0,
  "infra_retries": 0,
  "next_attempt_at": 1788942000.0,
  "frequency": {"held": 0, "blocked": 0, "by_governing_cap": {}},
  "rate": {"target_per_minute": 10, "actual_per_minute": 4}
}
```

`by_state` отдаёт собственные названия состояний платформы — без слоя перевода, потому что разница между «дозвонились» и «попытки кончились» важна именно вам. **Состояния с нулевым количеством в ответ не попадают.**

Состояние	Что означает
`pending`	ждёт набора
`in_progress`	набирается прямо сейчас
`awaiting`	набран, ждёт вердикта анализа разговора (стратегия `expert`)
`done`	решение окончательное — новых попыток не будет: дозвонились, или исход звонка не допускает повтора (номер не существует, блокировка спам-фильтром, в США — отклонённый абонентом вызов)
`exhausted`	попытки прекращены без разговора: израсходован бюджет попыток или инфраструктурных повторов, следующий повтор пришёлся бы позже окончания кампании, или кампания закончилась, пока контакт ждал (тогда ему не звонили ни разу)
`suppressed`	исключён до набора (реестр «не звонить», география, политика)

`rate.target_per_minute` — целевой темп, `rate.actual_per_minute` — сколько наборов ушло за последнюю минуту. У кампании с авто-скоростью в `rate` ещё:

Поле	Что означает
`auto`	`true` — кампания на авто-скорости.
`source`	`worker` — цель держит диалер после пересчёта; `estimate` — оценка, пока пересчёта не было.
`remaining_dials`	Сколько наборов ещё нужно, с учётом перезвонов.
`open_minutes`	Рабочие минуты до `ends_at`.
`cap_per_minute`	Лимит звонков в минуту сабгруппы кампании, `null` — лимита нет.
`capped`	`true` — чтобы успеть, нужно больше, чем позволяют лимиты: часть базы до `ends_at` не будет дозвонена.
`concurrency_slots`, `in_flight`, `talking`	При лимите одновременных звонков организации — сколько звонков допустимо, сколько в полёте и сколько из них разговаривают.
`ready_now`	Сколько открытых контактов можно набрать прямо сейчас — у остальных ещё не прошёл интервал перезвона.

Поле	Что означает
`interval_bound`	`true` — кампания набирает ниже цели не из-за лимитов: нужную скорость не из кого набрать, остальные контакты ждут интервал перезвона. Скорость при этом не меняется.

Остальные ключи `rate` — служебные входы оценки, их набор может меняться.

`ready` — сколько ожидающих контактов набираемы прямо сейчас; `holds` — сколько ожидающих удерживается (причины в `held_reasons`); `suppressed_reasons` раскладывает состояние `suppressed`; `dnc_suppressed_at_launch` + `dnc_suppressed_mid_campaign` = `suppressed_reasons.dnc_suppressed`; `frequency` сообщает, сколько контактов сейчас удержано или заблокировано ограничениями частоты на номер и какие лимиты этим управляют. Ключи причин внутри `held_reasons`, `suppressed_reasons` и `by_governing_cap` — машинные коды, их набор может расти.

> Обычная кампания закрывается сама, когда очередь опустела, — кампания, созданная с `accepts_call_additions: true`, не закрывается (её закрывают только `ends_at` или отмена). Отдельного вебхука «кампания завершилась» нет в любом случае — опрашивайте `progress` или `GET /v1/campaigns/{id}`.

Журнал пересчётов авто-скорости

`GET /v1/campaigns/{id}/rate-history` — по записи на каждый пересчёт авто-скорости, от старых к новым:

```
{
  "campaign_id": "cam-7f3a",
  "entries": [
    {"id": 412, "recorded_at": 1793605200.0, "rate": 24, "estimate": {"…": "…"}}
  ]
}
```

`rate` — темп, с которым кампания набирала после этого пересчёта; `estimate` — входы, из которых он получен: сколько наборов ещё нужно, рабочие минуты, доля перезвонов, потолки сабгруппы и одновременных звонков, звонки в полёте. Состав `estimate` служебный и может меняться. Листание — `?after=<id последней прочитанной записи>` и `?limit=` (по умолчанию 1000, не больше 5000). У кампании без авто-скорости список пуст; чужая или несуществующая кампания — `404 campaign_not_found`.

Снятие контакта с обзвона

Остановить набор номера до того, как звонок состоится: долг погашен, номер принадлежит другому человеку, вопрос закрыли в другом канале.

Метод и путь	Тело	Область
`POST /v1/campaigns/{id}/contacts/withdraw`	`phones` **или** `contact_ids`	эта кампания в любом статусе, кроме `completed` и `cancelled` (черновик — можно)
`POST /v1/contacts/withdraw`	`phones`	все **живые** кампании организации — `scheduled`, `running`, `paused`

```
curl -s -X POST "$BASE_URL/v1/contacts/withdraw" \
  -H "Authorization: Bearer $CFLOW_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{"phones": ["+7 701 123 45 67", "77015550011"]}'
```

`200 OK` раскладывает контакты по исходу; все пять ключей присутствуют всегда:

```
{
  "operation_id": "wd-3f9c2a71b0",
  "withdrawn":         [{"contact_id": "cc-…", "campaign_id": "cam-…", "phone": "7011234567"}],
  "already_withdrawn": [],
  "not_withdrawable":  [{"contact_id": "cc-…", "campaign_id": "cam-…", "phone": "7015550011", "reason":
```

```
"in_call"}],
  "not_found": [],
  "invalid":   []
}
```

- **Что снимается.** Контакт, ожидающий набора (`pending` , включая запланированный перезвон) или ожидающий вердикта аналитики (`awaiting`). Он переходит в `suppressed` с причиной `withdrawn_by_client` и в этой кампании больше не набирается. У контакта в `awaiting` звонок **уже состоялся** — снятие отменяет перезвон, который мог бы назначить вердикт.
- **Что не снимается.** `not_withdrawable` называет причину: `in_call` — звонок идёт или контакт только что взят в набор (снятие трубку не кладёт: используйте `POST /v1/calls/{call_id}/end` , а после звонка снимите контакт ещё раз, если перезвон не нужен), `done` , `exhausted` или `already_suppressed` — исключён раньше самой платформой (реестр «не звонить», частотный лимит, география); прежняя причина сохраняется.
- **Повторять безопасно.** `Idempotency-Key` не нужен: уже снятый контакт вернётся в `already_withdrawn` .
- **Как адресовать.** Номер сопоставляется по **последним десяти цифрам**, поэтому написание значения не имеет, а само значение можно слать строкой или числом. Один номер достаёт все контакты, которые его несут, — в кампании с дублями все сразу. Идентификатор контакта — это `metadata.contact_id` из вебхуков звонка; контакт, который ещё ни разу не набирали, адресуется номером.
- **Ручка по организации достаёт и кабинетные кампании**: все живые кампании организации, как бы они ни были созданы. Черновики и закрытые кампании не трогаются.
- **Частичная обработка.** Один плохой адрес не роняет запрос: в `invalid` вернётся то, что не удалось прочитать (не номер, нет цифр, пустой идентификатор), в `not_found` — то, под что не нашлось контакта, и то и другое ровно в том виде, в каком вы прислали. По умолчанию не больше 1000 адресов на запрос.
- **Последствия.** Если снять последние ожидающие контакты идущей кампании, она завершится на следующем проходе диалера — кроме кампании, открытой к добавлению номеров. Снятый номер вернётся в набор только с новой партией звонков.

> Экрана снятия в кабинете пока нет: причина `withdrawn_by_client` видна в разбивке прогресса кампании, а сама операция доступна только через API.

ТЕХНИЧЕСКИЙ СПРАВОЧНИК

Отчёт по загрузке номеров

Что вернулось по каждой строке файла и построчные коды замечаний.

Метод и путь	Что отдаёт
`GET /v1/imports/{import_id}`	сводку по загрузке (тот же объект, что `import` в ответе на создание) плюс живую/зафиксированную `dnc_screen`
`GET /v1/imports/{import_id}/report?format=csv`	отчёт по строкам: `csv` (по умолчанию) или `xlsx` (иное — `422 report_format_invalid`); имя файла `import-<id>-report.<ext>`
`GET /v1/imports/{import_id}/report?only=dnc_match&format=csv`	только точные совпадения с реестром; имя файла `import-<id>-dnc.<ext>`. Любое другое значение `only` — `422 report_filter_invalid`; загрузка без записанной проекции — `409 dnc_screen_required`

Загрузки видны только через кампании вашей организации (иначе `404 import_not_found`).

Объект сводки по загрузке

Поле	Тип	Что означает
`import_id`	string	
`status`	string	`accepted` | `refused`.
`rows_seen`, `accepted`, `warned`, `rejected`, `duplicates_found`	integer	Счётчики строк. У отклонённой партии добавляемых звонков `rejected` / `warned` считаются из замечаний.
`dnc_matched`	integer	`matched_rows` сверки (`0`, когда она недоступна).
`refusal_code`	string	`""` у принятой загрузки; иначе одно из `unreadable_file`, `no_rows`, `no_phone_column`, `ambiguous_phone_column`, `too_many_rows`, `no_usable_rows`, `missing_required_column`, `calls_rejected`, `calls_dnc_matched` (файловые коды вроде `unsupported_format`, `file_too_large` на `/v1` не встречаются).
`issues`	object	**С ключами по коду замечания**: `{"<код>": {"severity": "error"|"warning", "count": n, "rows": [номера строк, первые 10]}}`. Пустой объект, когда замечаний нет.
`dnc_screen`, `dnc_matches`		См. Поля кампании.

> `issues` в сводке — **объект с ключами по коду**, а не массив. Массив (элементы `{"row_number", "code", "severity"}`, первые 50) приходит только внутри `error.import` на отклонённой партии.

Построчные коды замечаний

Колонка `code` в отчёте и ключи `issues`:

Код	Серьёзность	Что означает
`phone_unreadable`	error	Нет читаемого телефона (меньше десяти цифр).
`row_empty`	error	Пустая строка.
`duplicate_row`	error	Дубль номера при `duplicate_action: "reject"`.
`duplicate_ignored`	warning	Дубль пропущен или заменён (`keep_first` / `keep_last`).

Код	Серьёзность	Что означает
`timezone_invalid`	warning	Неизвестное значение `timezone` (используется зона кампании).
`timezone_required`	error	Только партия `calls`: нет `timezone` в кампании с зоной контакта под профилем kz.
`group_role_invalid`, `group_key_missing`, `group_key_too_long`	error	Колонки договорных групп несогласованы.
`variable_name_collision`	warning	Две колонки отображаются в одну переменную.
`dnc_match`	warning	Номер есть в реестре «не звонить».
`state_mismatch`, `state_invalid`, `zip_invalid`, `city_mismatch`, `timezone_override`, `state_only`, `multi_zone`	warning	Пакет рынка США: флаги разрешения локации.
`unresolvable_location`, `non_callable_geography`	error	Пакет рынка США: строку набрать нельзя.

Файл отчёта

Колонки отчёта: `row_number`, `severity`, `code`, `supplied`, `duplicate_of_row` (`supplied` — JSON-объект с присланными значениями). Колонки с текстом сообщения **нет**: машиночитаемый контракт отчёта — это `code`, а превращать его в человеческий текст — задача потребителя.

Итоговый блок в конце файла (`import_summary`) подписан именами полей — `filename, format, encoding, phone_column, rows_seen, accepted, warned, rejected, duplicates_found, status` плюс `refusal` с кодом отказа у отклонённой загрузки; для источников `planned_calls` и `calls` поля `filename` и `encoding` пусты, а `format` равен `json`. У отчёта `dnc_match` колонки `row_number, phone, contact_name, phone_key, dnc_entry_id, source, reason, recorded_at` и блок `dnc_screen_summary` (`checked_rows, unscreened_rows, matched_rows, suppressed_contacts`).

Два соглашения общего механизма загрузок действуют на всех путях одинаково — и для файла в кабинете, и для `planned_calls` при создании кампании, и для добавляемой партии `calls`:

- **нумерация строк считает заголовок первой строкой**, поэтому первый planned call — это `row_number` 2, а N-й — N+1: и в `issues`, и в `error.import`, и в файле отчёта;
- **CSV — это UTF-8 с BOM и переводами строк CRLF** (чтобы Excel корректно открывал кириллицу): декодируйте как `utf-8-sig`, иначе первая ячейка заголовка прочитается как `row_number`.

Итоговые строки вместо числового `row_number` несут имена полей и строками данных не являются.

ТЕХНИЧЕСКИЙ СПРАВОЧНИК

Отдельные звонки

`POST /v1/calls/phone`, объект звонка, чтение, завершение и список звонков.

`POST /v1/calls/phone`

Один исходящий звонок: платформа регистрирует звонок, сразу отвечает и набирает номер из устойчивой очереди набора через указанный транк, подключая опубликованного агента, когда абонент ответит.

Поле	Тип	Обяз.	По умолч.	Описание
`agent_id`	string	да	—	Имя агента. Должен быть **опубликован** и перечислен в `agents` используемой исходящей группы.
`trunk_id`	string	да	—	Транк, через который набирать.
`to_number`	string	да	—	Номер назначения; нормализуется перед набором.
`route_id`	string	нет	`""`	Исходящая группа транка. Пусто → разрешается из пары (`trunk_id`, `agent_id`): агент, привязанный ровно к одной группе транка, выбирает её; несколько → `409 agent_subgroup_ambiguous`; ни одной → `403 agent_not_in_subgroup`. Неизвестный `route_id` → `404 subgroup_not_found`.
`from_number`	string	нет	`""`	Номер, с которого звоним; должен быть одним из `caller_numbers` **группы**. Пустая строка отдаёт выбор номера группе — по её политике исходящего номера: `random` (по умолчанию) берёт случайный, `round_robin` обходит пул по кругу. Кандидаты — пул группы, пересечённый с пулом гранта организации; остался один номер — он и будет выбран. Группа с пустым пулом отклоняет любое значение, включая пустое (`caller_number_not_allowed`); пустое пересечение с грантом — `caller_number_outside_grant_pool`.
`dynamic_variables`	object	нет	`{}`	Переменные, передаваемые в сценарий агента (имя клиента, данные счёта).
`metadata`	object	нет	`{}`	Ваша непрозрачная нагрузка для сопоставления. Возвращается без изменений в объекте звонка и в **каждом** вебхуке — кладите сюда идентификатор лида.
`ring_duration`	number | null	нет	`null`	Ограничение времени дозвона в секундах; должно быть > 0. `null` — значение агента по умолчанию.

Нормализация номера. Из `to_number` убираются все нецифровые символы (`"+1 (555) 123-4567"` → `"15551234567"`). Номер без цифр отклоняется с `400`. Если **группа** задаёт префикс набора, он добавляется к нормализованному номеру; разрешённые направления проверяются по нормализованному номеру **до** префикса.

Порядок проверок. Потолок очереди набора (`429 dial_queue_full`) проверяется первым, до всякой валидации. Затем синхронно и до регистрации звонка: остановка набора платформой (409), транк и грант (404), группа и привязка агента (404/409/403), номер звонящего (400), политика направлений (403), лимиты группы (429), `ring_duration` (400) и в конце — опубликованная версия агента (409). Отказ на любом из этих шагов не оставляет звонка. Поскольку лимиты проверяются до проверки публикации, `429` не говорит вам, опубликован ли агент.

Заголовок `Idempotency-Key` здесь рекомендуется (и фактически обязателен после `429`): повтор с тем же ключом возвращает **уже созданный звонок** вместо второго набора. Ключи помнятся сутки и ограничены вашим API-ключом и эндпоинтом. Переигровка возвращает объект звонка **в том виде, в котором он был при создании** (`status: "queued"`), а не его текущее состояние — для этого опрашивайте `GET /v1/calls/{call_id}`.

> **Успех — `202 Accepted`, а не `201`.** Звонок зарегистрирован и поставлен в очередь набора; сам набор происходит асинхронно после ответа. Сохраните `call_id`: это ключ сопоставления для опроса и для каждого вебхука.

Ошибки инициации звонка

Статус	`error.code`	Условие
400	`caller_number_not_allowed`	`from_number` не из номеров группы
400	`caller_number_outside_grant_pool`	`from_number` вне пула, выданного вашей организации (только транки по гранту)
400	`destination_number_invalid`	в `to_number` нет цифр
400	`ring_duration_invalid`	`ring_duration` ≤ 0
403	`agent_not_in_subgroup`	агент не привязан к группе (это же ответ и на несуществующее имя агента)
403	`destination_not_allowed`	направление заблокировано политикой группы
404	`trunk_not_found` / `subgroup_not_found`	неизвестный `trunk_id` (или нет гранта на него) / неизвестный `route_id`
409	`agent_subgroup_ambiguous`	агент в нескольких группах транка — пришлите `route_id`
409	`agent_not_published`	у агента нет опубликованной версии
409	`dialing_stopped`	исходящий набор вашей организации остановлен платформой. Звонки, принятые в очередь до остановки, не теряются: они ждут, пока остановку снимут, и набираются после этого. Снять остановку может только платформа — обратитесь в поддержку
422	`invalid_request`	тело не проходит проверку схемы; детали по полям — в `message`
502	`dial_failed`	только если оператор отключил очередь набора (синхронный запасной режим). В обычном асинхронном режиме сбой набора — **не** HTTP-ошибка: звонок создан (`202`) и позже читается как `status: "failed"` , `status_reason: "dial_failed"`

Объект звонка

Приходит в `POST /v1/calls/phone` , `GET /v1/calls/{call_id}` и `POST /v1/calls/{call_id}/end` .

Поле	Тип	Описание
`call_id`	string	Идентификатор звонка: признак плюс полный UUID — `call-out-<uuid>` исходящий, `call-in-<uuid>` входящий, `web-<uuid>` веб. UUID — 36 строчных символов с дефисами, например `call-out-0f8fad5b-d9cb-469f-a165-70867728950e` . Номера телефона и других данных в нём нет.
`agent_id`	string	Имя агента.
`flow_version`	string | number | null	Отметка версии опубликованного сценария, на котором шёл звонок. **Тип не стабилен**: пока звонок живой — непрозрачный отпечаток-строка; после сохранения — целочисленный номер версии. Считайте непрозрачным, а для идентичности используйте `flow_version_id` .
`flow_version_id`	number | null	Числовой идентификатор опубликованной версии.
`channel`	string	`phone` | `web` .
`direction`	string	`outbound` | `inbound` | `web` .
`status`	string	Живое состояние: `queued` → `ringing` → `ongoing` → `ended` | `failed` .
`call_status`	string	Финальная корзина: `completed` | `not_connected` | `error` ; `""` до финализации.
`status_reason`	string	Уточнённая причина внутри корзины; `""` до финализации.
`outcome_category`	string	Строка разбивки исходов кампании, в которую попал звонок: `talked` («Разговор с человеком»), `transferred` , `screener_human_handoff` , `voicemail` , `screener_voicemail_handoff` , `screener_no_handoff` , `connected_no_conversation` и другие; `""` до финализации. Есть и в элементах списка `GET /v1/calls` .

Поле	Тип	Описание
`error_code`	string | null	Код ошибки оператора связи или телефонии, если платформа его получила; иначе `null`. Диагностическое.
`message_left`	boolean	Сработало определение автоответчика, и агент оставил сообщение — прозвучавшее целиком.
`callee_reached`	boolean	Набор дошёл до телефона абонента: были гудки или ранний звук. У недозвона отличает «звонило, не взяли» от «даже не зазвонило». У входящих и веб-звонков — `false`.
`voicemail_decision`	object | null	Решение по автоответчику для звонка кампании: `mode` — режим политики (`always` | `rules` | `never`), `source` — откуда политика (`campaign_override` | `organization_policy` | `unconfigured`), `policy` — сама политика, `blocked_by` — какое условие запретило сообщение (`""`, если не запрещало), `counters` — `voicemail_today` и `voicemail_7d` по номеру, `outcome` — итог: `left` (сообщение оставлено) | `not_left` (автоответчик, сообщение не прозвучало целиком) | `suppressed` (политика запретила) | `no_voicemail` (автоответчика не было), `message` — чем кончилось само сообщение: `left` (прозвучало до конца) | `cut` (оборвано на середине) | `unvoiced` (так и не прозвучало) | `closed` (автоответчик закрылся до сообщения) | `error` (сбой сообщения), `null`, если агент не пытался его оставить; счётчики считают только оставленные сообщения. `null` у звонка не из кампании, в том числе начатого через `/v1`. См. «Политика голосовых сообщений».
`screener`	object | null	Фаза фильтра звонков: `detected` — `true`; `type` — какой фильтр (`ios` | `assistant` | `google` | `cloaked` | `unknown`); `outcome` — чем кончилась фаза (`connected` — соединил с человеком, `message` — предложил оставить сообщение, `declined` — отклонил, `timeout` — истекло время); `replies` — сколько раз агент ответил фильтру; `hold_s` — сколько секунд фильтр держал линию; `source` — как фильтр узнан (`phrase` | `model` | `watch`); `greeting` — первые слова фильтра. `null`, если фильтра в звонке не было.
`started_at`	number | null	Начало попытки (для телефонных звонков — до набора).
`answered_at`	number | null	Момент, когда абонент ответил; `null` для недозвонов и веб-звонков.
`media_started_at`	number | null	Когда пошёл аудиопоток. Диагностическое.
`talk_from`, `talk_to`	number | null	Границы интервала разговора, по которому считается `duration`. Диагностическое.
`ended_at`	number | null	Конец звонка; `null`, пока звонок живой.
`duration`	number | null	Время разговора в секундах от `answered_at`. `null` для недозвонов, для `error/not_finalized` и пока идёт дозвон; у живого звонка — прошедшее время разговора.
`transcript`	array	Упорядоченные реплики: `{"role": "user"|"assistant", "text": string, "interrupted"?: true, "out_of_turn"?: true, "wait_request"?: true, "voicemail_greeting"?: true, "ts"?: number, "spoke_at"?: number}`. В транскрипт входит только то, что абонент действительно услышал: реплика агента, которая так и не прозвучала (звонок закончился или её оборвали до первого звука), в транскрипт **не попадает** вовсе — её видно только в карточке звонка с «Показать системные события», с пометкой «не прозвучало». С 14.10.2026 так же и для старых звонков при повторной отправке вебхука. Реплика, оборванная на середине, остаётся — с усечённым текстом и `interrupted: true`. `voicemail_greeting: true` (только у реплик абонента) — приветствие автоответчика, по которому звонок определён как автоответчик; агент на него не отвечал. `out_of_turn: true` (только у реплик абонента) — абонент говорил поверх непрерываемой реплики агента; реплика записана вне хода и не является ответом на следующий вопрос агента. `wait_request: true` (только у реплик абонента) — абонент попросил агента подождать, и агент включил паузу (пауза по просьбе клиента); у такой реплики стоит и `out_of_turn: true`. Ключи добавлены без смены версии: разборщики, которые о них не знают, продолжают работать.

Поле	Тип	Описание
`dynamic_variables`	object	Текущие переменные разговора: ваши начальные значения плюс собранные во время звонка. На **недозвоне** здесь лежат значения, переданные вами для контакта: пустой объект не является признаком недозвона — для этого читайте `call_status`.
`metadata`	object	Ваша нагрузка, без изменений. У звонка кампании вместо неё — ключи кампании и контакта, см. «Звонки кампании».
`analysis_status`	string	`""` (не начат) | `pending` | `processing` | `done` | `skipped` | `failed` | `interrupted`. Терминальные: `done`, `skipped`, `failed`, `interrupted`.
`call_analysis`	object | null	Только когда `analysis_status` равен `done`: `summary`, `sentiment`, `custom_fields`, `call_successful`.
`health`	string | null	Техническое здоровье звонка: `ok` | `warning` | `error`; `null` до финализации. Отражает проблемы провайдеров и инфраструктуры, **а не** качество разговора.
`health_reasons`	array	Коды причин вердикта, см. глоссарий ниже.
`events`	array	События звонка без деталей: `{"type", "turn", "ts", "title"?, "status"?}` (инциденты наблюдателя, переводы и подобные моменты). Дополнительное, диагностическое.
`cost`	object | null	`{"total_microusd", "analytics_microusd"}` — в ответах вашего ключа `null`.
`recording_url`	string	Устойчивый адрес скачивания, когда запись готова; до этого `""`. Запрос — с API-ключом организации звонка (`Authorization: Bearer <API-ключ>`).
`recording`	object | null	Метаданные записи: `{"id", "status", "channels", "format", "duration", "size", "expires_at", "client_audio_lost_sec"}`. `status` — `recording` | `processing` | `ready` | `failed` | `deleted`; `channels` — `stereo` | `mono`; `client_audio_lost_sec` — сколько секунд посреди разговора в записи нет звука клиента, `null` — не измерялось (см. Вебхуки).
`from_number`, `to_number`, `trunk_id`, `route_id`	string	Только телефонные звонки.

Жизненный цикл: `queued` (набор инициирован, дозвона ещё нет) → `ringing` → `ongoing` (абонент ответил) → `ended` (разговор состоялся, включая сообщения на автоответчик и переводы) либо `failed`.

> `failed` **сам по себе не означает, что абонент не ответил.** Обычно он приходит с `call_status: "not_connected"` (`no_answer`, `busy`, `dial_failed`, ...), но сбой платформы тоже читается как `failed` с `call_status: "error"` (`internal_error`, `not_finalized`, `agent_unavailable`, `no_media`). Всегда читайте `call_status` / `status_reason`.

Веб-звонки пропускают `ringing` и `failed`: они идут `queued` → `ongoing` (первая реплика истории) → `ended`.

Асинхронные поля заполняются после завершения звонка: `call_analysis` — когда закончится пост-анализ, `recording_url` — когда финализируется запись. Вебхук `call_analyzed` сообщает, когда готово и то, и другое, — опрос это запасной путь, а не основной механизм.

Чтение и завершение звонка

`GET /v1/calls/{call_id}` возвращает полный объект звонка; на неизвестный идентификатор — `404 call_not_found`.

Звонки, записанные до формата с UUID. Их старый идентификатор был с коротким хвостом (`call-out-<16 hex>`, `web-<16 hex>`), а у входящего в нём был номер звонящего. Теперь такие звонки отвечают псевдонимом в новом формате (`<признак><uuid>`, признак тот же) в поле `call_id` — во всех ответах и вебхуках, — а старый идентификатор по-прежнему принимается на вход: `GET /v1/calls/{старый ID}` отвечает `200` с `call_id`, равным

псевдониму. Если вы сравниваете `call_id` из ответа с сохранённым старым идентификатором как строки, расхождение будет только у таких звонков; `event_id` их вебхуков не меняется.

`POST /v1/calls/{call_id}/end` завершает живой звонок — телефонный звонок, ещё стоящий в очереди набора, отменяется до набора — и возвращает объект звонка.

> `end` идемпотентен только **внутри окна живого реестра**: вызов на звонке, который закончился менее 5 минут назад, вернёт объект без ошибки; после этого запись в реестре исчезает и эндпоинт отвечает `404 call_not_found`, хотя `GET /v1/calls/{call_id}` по-прежнему возвращает сохранённый объект. Не используйте `end` как способ прочитать состояние.

`GET /v1/calls/{call_id}/turns` отдаёт метрики задержек по тёрнам завершённого телефонного или веб-звонка: массив `{"turn", "metrics": {"vad_ms", "asr_ms", "eou_wait_ms", "classifier_ms", "llm_ms", "tts_ms", "turn_ms", …, "estimated"?: true}, "interrupted"?: true, "merged_from"?: number}`. Диагностическая поверхность для нагрузочных тестов; `404` на неизвестные идентификаторы и на сессии, не являющиеся звонком.

`merged_from` появляется у тёрна абонента, чью фразу расколола пауза и которую склеили обратно: это число обрывков, а фазы тёрна взяты у первого из них. Ключ аддитивный: клиент, который его не читает, разбирает ответ обычным образом. Метрики `llm_first_sentence_ms` в ответе нет.

`GET /v1/calls` — список звонков

Параметр	Примечания
`agent_id`	фильтр по агенту
`direction`	`outbound` | `inbound` | `web`. **Не валидируется**: неизвестное значение даёт пустой список, а не `400`
`status`	только `ongoing` или `ended` (иное → `400 invalid_filter_value`); `ended` включает неудавшиеся звонки
`call_status`	`completed` | `not_connected` | `error` (иное → `400 invalid_filter_value`)
`status_reason`	точное совпадение, например `agent_hangup` (неизвестное значение → `400 invalid_filter_value`)
`health`	`ok` | `warning` | `error` — «только проблемные звонки» одним кликом: `health=error`
`health_reason`	точное совпадение кода. Не валидируется: неизвестный код даёт пустой список
`from_number`, `to_number`	⚠ применяются **после** постраничности, только к текущей странице
`since`, `until`	границы по началу звонка, секунды эпохи, включительно
`limit`	1–200, по умолчанию 50; вне диапазона → `422 invalid_request`
`cursor`	курсор из предыдущего ответа; повреждённый → `400 invalid_cursor`

Ответ: `{calls: [...], has_more, next_cursor}`. Элементы списка — сводки без транскрипта; полный объект — `GET /v1/calls/{call_id}`. Идите по страницам, передавая `next_cursor` как `cursor`, пока `has_more` не станет `false`. В списке только телефонные и веб-звонки; тестовые сессии редактора и чаты в него не попадают.

> `from_number` / `to_number` — фильтры уровня страницы: страница может содержать меньше `limit` элементов (в том числе ноль), пока `has_more` всё ещё `true`. Не считайте пустую страницу концом результатов. Для сверки по номерам лучше фильтровать по своей `metadata` на своей стороне.

ТЕХНИЧЕСКИЙ СПРАВОЧНИК

Звонок с сайта: Web SDK

`POST /v1/calls/web`, браузерный SDK `@hubtalk-ai/web-sdk` и готовый виджет: разговор с агентом прямо на вашем сайте.

Посетитель вашего сайта нажимает кнопку и говорит с агентом из браузера — без телефонной линии, по микрофону. Для этого есть эндпоинт `POST /v1/calls/web` и браузерный SDK `@hubtalk-ai/web-sdk`, а в SDK — готовый виджет: кнопка в углу страницы и панель звонка.

Если нужно просто дать человеку поговорить с агентом, ничего не встраивая, хватит демо-ссылки. Web SDK — для случая, когда звонок начинается на вашем сайте, на вашем домене и с переменными, которые знает ваша система.

Как это устроено

1. Посетитель нажимает кнопку звонка на вашей странице.
2. Страница просит **ваш сервер** начать звонок. Сервер вызывает `POST /v1/calls/web` с API-ключом организации и получает сессию: `access_token` и `url`.
3. Сервер отдаёт сессию странице, страница передаёт её в SDK. Дальше разговор идёт напрямую между браузером и платформой, а состояние звонка приходит в страницу событиями.

> **API-ключ не попадает в браузер.** Его знает только ваш сервер. Странице достаётся одноразовая сессия на один звонок — она даёт войти только в этот разговор и больше ничего.

Что нужно заранее

- **Агент опубликован.** Звонит опубликованная версия; черновик не подключится. Привязка к телефонии для web-звонка не нужна.
- **Выпущен API-ключ** — раздел **«API-ключи»**, см. Ключ доступа. Ключ хранится на вашем сервере.
- **Страница открыта по HTTPS.** Без защищённого соединения браузер не даст доступ к микрофону. Исключение — `localhost` при разработке.

`POST /v1/calls/web`

Создаёт web-звонок и выдаёт сессию для браузера. Запрос делает ваш сервер, с заголовком `Authorization: Bearer <API-ключ>`.

Поле	Тип	Обяз.	По умолч.	Описание
`agent_id`	string	да	—	Имя агента. Должен быть **опубликован**.
`dynamic_variables`	object	нет	`{}`	Переменные сценария — те же `{{var}}`, что в узлах: имя клиента, номер заказа.
`metadata`	object	нет	`{}`	Ваша непрозрачная нагрузка для сопоставления. Возвращается без изменений в объекте звонка и в каждом вебхуке.
`avatar`	boolean	нет	`false`	Попросить живой аватар для этого звонка, см. Живой аватар. Только `true` или `false`: строка или число — `422`.
`avatar_quality`	string	нет	настройка агента	Качество видео аватара: `low`, `medium`, `high`, `very_high`. Неизвестное значение не ломает звонок: платформа его пропускает и берёт качество из настройки агента.

```
curl -X POST "$BASE_URL/v1/calls/web" \
  -H "Authorization: Bearer $HUBTALK_API_KEY" \
  -H "Content-Type: application/json" \
  -H "Idempotency-Key: 6f1c2a90-site-call-1" \
```

```
  -d '{
    "agent_id": "clinic_booking",
    "dynamic_variables": {"client_name": "Айгерим"},
    "metadata": {"lead_id": "L-1042"}
  }'
```

Ответ — `201 Created`: объект звонка и три поля сессии.

Поле	Описание
`access_token`	Токен для браузера. Живёт **2 часа** и не продлевается.
`url`	Адрес, к которому подключается браузер.
`room`	Комната звонка; совпадает с `call_id`.
`call_id`	Идентификатор звонка вида `web-<uuid>`. По нему звонок читается через `GET /v1/calls/{call_id}` и находится в вебхуках.

Ответ целиком можно отдать странице и передать в SDK как есть: лишние поля SDK пропускает.

Ошибки

Статус	`error.code`	Условие
404	`agent_not_found`	агента с таким именем нет в вашей организации
409	`agent_not_published`	у агента нет опубликованной версии
422	`invalid_request`	тело не проходит проверку схемы; детали по полям — в `message`
429	`org_concurrency_limit`	достигнут лимит одновременных звонков организации
503	`agent_dispatch_failed`	платформе сейчас некуда подключить агента — повторите позже

Общие ошибки ключа и лимита запросов — в «Ошибках».

Повтор запроса. Заголовок `Idempotency-Key` защищает от двойного звонка при ретраях: повтор с тем же ключом в течение суток вернёт ту же сессию, второго звонка не будет. Ключ повтора — на один звонок: для следующего посетителя нужен новый.

Сессия — на один звонок. Токен живёт 2 часа и не продлевается, поэтому выдавайте новую сессию на каждый звонок, а не держите одну на открытой странице.

Звонок в Истории и вебхуках

Web-звонок сразу появляется в Истории как активный разговор канала **«Web-звонок»** (`channel: "web"`, `direction: "web"`). Транскрипт, запись, пост-аналитика и вебхуки работают как у телефонного звонка. В сценарии переменная `{{direction}}` равна `web` — по ней агент может говорить с посетителем сайта иначе, чем по телефону.

Жизненный цикл короче телефонного: `queued` → `ongoing` → `ended`, без `ringing` и `failed`. Поля номеров и транка у web-звонка пустые, `answered_at` — `null`.

Установка SDK

```
npm install @hubtalk-ai/web-sdk
```

```
import { VoiceClient } from '@hubtalk-ai/web-sdk';
```

Без сборщика. В пакете, в папке `dist`, лежат готовые файлы для подключения обычным тегом `<script>`: `voice-client.bundle.js` — клиент, `widget.bundle.js` и `widget.css` — виджет. Положите их на свой сервер:

```
<script src="/sdk/voice-client.bundle.js"></script>
<script>
  const client = new VoiceClient();
</script>
```

Сторонний CDN не нужен: всё, что требуется SDK, внутри этих файлов.

Первый звонок

На сервере — эндпоинт, который вызывает `POST /v1/calls/web` и возвращает ответ странице. Пример на Node.js (Express):

```
app.post('/api/start-voice-call', async (req, res) => {
  const r = await fetch(`${process.env.HUBTALK_BASE_URL}/v1/calls/web`, {
    method: 'POST',
    headers: {
      Authorization: `Bearer ${process.env.HUBTALK_API_KEY}`,
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({
      agent_id: 'clinic_booking',
      dynamic_variables: { client_name: req.user?.name ?? '' },
    }),
  });
  res.status(r.status).json(await r.json());
});
```

Переменные звонка задаёт сервер. Если какое-то значение приходит со страницы, принимайте только те поля, которые вы сами ожидаете, — так посетитель не подставит в сценарий чужие данные.

На странице:

```
const client = new VoiceClient();

startButton.addEventListener('click', async () => {
  const session = await fetch('/api/start-voice-call', { method: 'POST' }).then((r) => r.json());
  await client.startCall(session);
});

client.on('transcript', ({ role, text, final, id }) => renderBubble(id, role, text, final));
client.on('callEnded', ({ reason }) => console.log('звонок завершён:', reason));

endButton.addEventListener('click', () => client.stopCall());
```

> `startCall` **вызывайте внутри обработчика нажатия.** Браузер разрешает звук агента и доступ к микрофону только после действия пользователя. В Safari на iPhone сетевой запрос между нажатием и `startCall()` может «съесть» это действие — тогда запрашивайте сессию заранее, например при открытии панели звонка.

VoiceClient

Параметры

`new VoiceClient(options?)` — все параметры необязательны.

Параметр	По умолчанию	Описание
`audioContainer`	скрытый `<div>`, который создаёт SDK	Куда SDK кладёт `<audio>` с голосом агента.

Параметр	По умолчанию	Описание
`logger`	`console`	Свой приёмник логов `{ error, debug }`. Токены в логах маскируются в любом случае.
`logLevel`	`'error'`	`'silent'`, `'error'` или `'debug'`.
`connectTimeoutMs`	15 секунд	Сколько ждать подключения.

Методы

Метод	Что делает
`startCall(session, options?)`	Подключается и включает микрофон. Принимает ответ `POST /v1/calls/web` как есть или `{ accessToken, url }`. Второй вызов во время звонка — ошибка `ALREADY_IN_CALL`. `{ audio: false }` — текстовый звонок.
`stopCall()`	Завершает звонок. Если звонка нет — ничего не делает, без ошибки.
`sendText(text)`	Отправляет агенту текст от посетителя: агент прерывает свою речь и отвечает, как на сказанное голосом. Вне звонка — `NOT_IN_CALL`; текст длиннее 1000 символов обрезается. В чат сообщение само не добавляется — оно вернётся событием `transcript` с `role: 'user'`.
`setMicrophoneEnabled(enabled)`	Выключает и включает микрофон во время голосового звонка; звонок при этом не прерывается. `isMicrophoneEnabled()` — текущее состояние.
`resumeAudio()`	Включает звук агента, если браузер его заблокировал (событие `audioPlaybackBlocked`). Вызывайте по нажатию кнопки «Включить звук».
`getState()`	Состояние соединения: `idle`, `connecting`, `connected`, `reconnecting`, `disconnected`, `error`.
`getAgentState()`	Что делает агент: `unknown`, `initializing`, `listening`, `thinking`, `speaking`.
`getCallId()`	`call_id` звонка — тот же, что принимает `GET /v1/calls/{call_id}`; до подключения — `null`.
`on` / `off` / `once`	Подписаться на событие, отписаться, подписаться один раз.
`destroy()`	Завершает звонок и освобождает всё; после этого клиент не используется.

События

Событие	Данные
`callStarted`	`{ callId }`
`callEnded`	`{ callId, reason }`. `reason`: `local` — завершили вы (`stopCall`), `remote` — завершил агент, `connection_lost` — пропала связь, `error` — сбой.
`connectionStateChanged`	`{ state, previous }`
`agentStateChanged`	`{ state, previous }`
`agentSpeakingStarted` / `agentSpeakingStopped`	`{}`
`transcript`	`{ role, text, final, id }` — реплика разговора, `role`: `user` или `agent`. Пока `final: false`, реплика ещё дописывается: рисуйте один пузырь на каждый `id` и обновляйте его.
`microphoneStateChanged`	`{ enabled }` — микрофон включён или выключен.
`audioPlaybackBlocked`	`{}` — браузер заблокировал звук агента. Звонок продолжается; покажите кнопку, которая вызовет `resumeAudio()`.
`error`	`VoiceError` — см. ниже.

Текстовый звонок

`startCall(session, { audio: false })` начинает тот же звонок без микрофона и звука: браузер не спрашивает разрешений, разговор идёт текстом через `sendText()` и события `transcript`. Это режим для посетителя, который не может или не хочет говорить вслух. Ответы агента приходят в темпе речи.

Ошибки SDK

Любой сбой — объект `VoiceError`: `{ code, message, state, cause? }`. Решения принимайте по `code`; `message` — строка для логов, текст для посетителя пишете вы.

Код	Когда
`INVALID_PARAMS`	в сессии нет `access_token` или `url`
`WEBRTC_UNSUPPORTED`	браузер не поддерживает звонки или страница открыта не по HTTPS
`MICROPHONE_PERMISSION_DENIED`	посетитель запретил доступ к микрофону
`MICROPHONE_UNAVAILABLE`	микрофона нет или он занят
`AUDIO_PLAYBACK_BLOCKED`	браузер повторно заблокировал звук после `resumeAudio()`
`TOKEN_EXPIRED`	сессия истекла до начала звонка
`TOKEN_INVALID`	платформа не приняла токен сессии
`CONNECTION_FAILED`	не удалось подключиться
`CONNECTION_LOST`	связь пропала и не восстановилась
`ALREADY_IN_CALL`	звонок уже идёт
`NOT_IN_CALL`	действие вне звонка: `sendText`, переключение микрофона
`SERVER_ERROR`	любой другой сбой

Готовый виджет

Если своя вёрстка не нужна, в пакете есть виджет: кнопка в углу страницы и панель звонка с состоянием агента, таймером, кнопками микрофона и завершения, чатом и транскриптом после звонка. Тексты виджета — на английском или русском.

```
import { mountWidget } from '@hubtalk-ai/web-sdk/widget';
import '@hubtalk-ai/web-sdk/widget.css';

const widget = mountWidget({
  container: document.getElementById('support'),
  locale: 'ru',
  async createSession() {
    const r = await fetch('/api/start-voice-call', { method: 'POST' });
    if (!r.ok) throw new Error('Session unavailable');
    return r.json(); // { access_token, url, … } — без API-ключа
  },
});

// при уходе со страницы или удалении компонента
widget.destroy();
```

Без сборщика — те же файлы из `dist`:

```
<link rel="stylesheet" href="/sdk/widget.css">
<div id="support"></div>
<script src="/sdk/widget.bundle.js"></script>
<script>
  HubTalkWidget.mountWidget({
```

```
    container: document.getElementById('support'),
    locale: 'ru',
    createSession: async () => {
      const r = await fetch('/api/start-voice-call', { method: 'POST' });
      if (!r.ok) throw new Error('Session unavailable');
      return r.json();
    },
  });
</script>
```

Параметр	По умолчанию	Описание
`container`	обязателен	Элемент страницы, в котором живёт виджет.
`createSession`	обязателен	Функция, которая просит ваш сервер начать звонок и возвращает ответ `POST /v1/calls/web`. Виджет вызывает её при нажатии **«Начать разговор»**.
`locale`	`'en'`	Язык текстов виджета: `'en'` или `'ru'`.
`mode`	`'standard'`	`'standard'` — панель звонка и чата; `'avatar'` — панель с карточкой живого аватара.
`screenAdaptation`	`{ mode: 'height' }`	Как открытая панель встаёт на экране. `'height'` — панель в углу, высотой в половину экрана. `'full-height'` — панель от шапки сайта до низа экрана; шапку виджет находит сам, или укажите её селектор в `headerSelector`. На телефонах панель занимает весь экран в любом режиме.
`renderCallDetails`	нет	Функция, которая один раз рисует в панели вашу форму — например, выбор темы обращения. Значения формы `createSession` читает сама. Во время звонка виджет форму прячет.
`avatar`	нет	Настройки карточки аватара для режима `'avatar'`, см. ниже.

Стили виджета подключаются отдельным файлом `widget.css` и действуют только внутри виджета — оформление вашей страницы они не меняют.

Живой аватар

В режиме `'avatar'` панель показывает **живое видео AI-аватара**: лицо, которое говорит голосом агента синхронно с речью. Видео идёт по тому же соединению, что и звук.

- **Аватар включается для агента поддержкой платформы** — она же задаёт, какое лицо будет в звонке. Напишите в поддержку, для какого агента нужен аватар, и уточните, какое лицо выбрано.
- **Ваш сервер просит аватар** полем `"avatar": true` в `POST /v1/calls/web`. Виджет подсказывает, когда это нужно: `createSession` получает `{ avatar, avatarQuality }`, и `avatar` равно `true` в режиме `'avatar'`. Это просьба, а не гарантия: если у агента аватар не включён, звонок идёт только со звуком, ответ API не меняется.
- **Заставку показываете вы.** Пока видео нет — до звонка, в первые секунды, если видео прервалось, — на месте аватара стоит ваша картинка `avatar.posterUrl` или пара беззвучных роликов. Подберите заставку, похожую на лицо аватара, иначе смена будет заметна.

```
mountWidget({
  container: document.getElementById('support'),
  mode: 'avatar',
  locale: 'ru',
  async createSession({ avatar, avatarQuality }) {
    const r = await fetch('/api/start-voice-call', {
      method: 'POST',
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify({ avatar, avatar_quality: avatarQuality }),
    });
    if (!r.ok) throw new Error('Session unavailable');
    return r.json();
  },
```

```
  avatar: {
    name: 'Алина',
    subtitle: 'Ассистент клиники',
    posterUrl: '/presenter/poster.jpg',
    liveCaptions: true,
  },
});
```

Параметр `avatar`	По умолчанию	Описание
`posterUrl`	обязателен	Картинка-заставка с вашего сервера.
`listeningVideoUrl`, `speakingVideoUrl`	нет	Беззвучные ролики-заставки «слушает» и «говорит» — оба или ни одного. Без них стоит картинка.
`name`, `subtitle`	«AI assistant», нет	Имя и подпись в шапке панели.
`quality`	настройка агента	`low`, `medium`, `high`, `very_high`. Уходит в `createSession` как `avatarQuality` — передайте его в `avatar_quality`.
`liveCaptions`	`false`	Последние реплики агента субтитрами под видео.

Пока идёт видео, на нём видна плашка **AI avatar**. Если в вашей стране посетителя нужно предупреждать, что он говорит с ИИ, — это остаётся на вашей стороне: плашка помогает, но не заменяет такое уведомление.

Без виджета то же видео доступно в `VoiceClient`: событие `agentVideoChanged` (`{ available }`) сообщает, появилось ли видео, а `attachAgentVideo(element)` выводит его в ваш `<video>`.

Требования к странице

- **HTTPS.** Без защищённого соединения браузер не даст микрофон, и `startCall` вернёт `WEBRTC_UNSUPPORTED`.
- **Content Security Policy.** Если у сайта есть CSP, разрешите: адрес, откуда грузятся файлы SDK, — в `script-src` и `style-src`; ваш сервер и адрес из поля `url` сессии (HTTPS и WSS) — в `connect-src`; `blob:` — в `media-src`. Для аватара: адрес картинки-заставки — в `img-src`, роликов — в `media-src`; с роликами ещё `style-src-attr 'unsafe-inline'`.
- **Браузеры.** Нужна поддержка WebRTC — она есть во всех современных браузерах на компьютере и телефоне.

ТЕХНИЧЕСКИЙ СПРАВОЧНИК

Исходы звонка: `call_status` и `status_reason`

Ось исходов: корзины, причины внутри них и как они складываются в вердикт.

`call_status` — финальная корзина завершённого звонка (пустая строка до финализации). Одна ось, один источник истины; качество (`call_successful`) и техническая надёжность (`health`) — отдельные оси и в статус не протекают:

Значение	Что означает
`completed`	звонок соединился и завершился (любым образом)
`not_connected`	исходящий звонок так и не соединился, или звонящий на входящем положил трубку до ответа агента
`error`	платформа не смогла провести звонок: соединённый звонок оборвался из-за сбоя платформы, агент так и не подключился, абонент ответил, но звук от него так и не пришёл, либо исход так и не был зафиксирован (детали — в `health`)

`status_reason` — уточнённая причина, всегда принадлежащая ровно одной корзине:

Колонка **«Анализ»** говорит, запускается ли по звонку пост-аналитика — саммари, извлечённые поля, оценка успешности. Это значение по умолчанию: список причин, по которым анализ пропускается, настраивается у агента.

Корзина	Значение	Что означает	Анализ
`completed`	`agent_hangup`	Разговор дошёл до конца сценария агента.	да
`completed`	`user_hangup`	Абонент говорил, но повесил трубку до конца сценария.	да
`completed`	`no_speech`	Звонок соединился, но распознаваемой речи абонента не было.	нет
`completed`	`too_short`	Ниже настроенного у агента минимума реплик или длительности. По умолчанию порог выключен, и эта причина не встречается.	нет
`completed`	`voicemail_reached`	Определён автоответчик (смотрите `message_left`).	только если сообщение оставлено
`completed`	`voicemail_suppressed`	Автоответчик определён, но правило ограничения сообщений запретило оставлять сообщение.	нет
`completed`	`max_duration_reached`	Автозавершение по максимальной длительности звонка у агента.	да
`completed`	`silence_timeout`	Автозавершение после долгого молчания абонента.	да
`completed`	`transferred_cold` / `transferred_warm`	Звонок успешно передан; ИИ вышел из разговора.	да
`completed`	`observer_incident`	Автозавершение реакцией наблюдателя соответствия.	да
`completed`	`platform_shutdown`	Платформа завершила звонок, останавливаясь на обновление.	да
`completed`	`screener_declined`	Ответил фильтр звонков, и ассистент абонента отклонил звонок или положил трубку, пока держал линию. Человек не говорил. Только у агентов с включённой обработкой фильтра звонков.	нет
`completed`	`screener_timeout`	Ответил фильтр звонков, и разговор с ним превысил таймаут удержания или таймаут фазы фильтра. Человек не говорил. Только с включённой обработкой фильтра звонков.	нет
`completed`	`ivr_gate`	Ответило голосовое меню с кнопками (IVR). Агент кнопки не нажимает и завершает звонок. Только с включённой обработкой фильтра звонков.	нет

Корзина	Значение	Что означает	Анализ
`not_connected`	`no_answer`	Телефон абонента зазвонил, и трубку не взяли за отведённое окно дозвона.	нет
`not_connected`	`busy`	Занято.	нет
`not_connected`	`user_declined`	Абонент явно отклонил звонок.	нет
`not_connected`	`invalid_destination`	Несуществующий или неверный номер.	нет
`not_connected`	`trunk_auth_failed`	Не прошла аутентификация SIP-транка.	нет
`not_connected`	`provider_unavailable`	Вызов до абонента не дошёл: телефония не ответила в срок или отказала, гудков не было.	нет
`not_connected`	`sip_routing_error`	Сбой маршрутизации SIP.	нет
`not_connected`	`spam_blocked`	Заблокировано оператором как спам.	нет
`not_connected`	`dial_failed`	Общий сбой набора (более точных деталей нет). Номер перезванивается через интервал кампании; в экспертной логике это условие **«Недозвон»**.	нет
`not_connected`	`cancelled_by_function`	Функция агента до набора решила звонок не размещать.	нет
`not_connected`	`caller_abandoned`	Входящий звонок: звонящий положил трубку до ответа агента. Разговора не было; `duration` равен `null`. До 14.10.2026 такие звонки приходили как `completed` / `no_speech`.	нет
`not_connected`	`cancelled_by_operator`	Набор остановлен командой извне до соединения — кнопкой кабинета или `POST /v1/calls/{id}/end`. Попытка контакта и частотный потолок на неё не тратятся.	нет
`error`	`internal_error`	Сбой платформы на соединённом звонке — смотрите `health` / `health_reasons`.	нет
`error`	`agent_unavailable`	Голосовой агент не подключился к звонку.	нет
`error`	`no_media`	Абонент ответил, но звук с его стороны так и не пришёл (оператор не передал медиа за весь звонок). `duration` настоящая; транскрипта нет. Только исходящие и только если у агента включена опция **«Звонок без медиа»** (по умолчанию включена); если её выключить, такой звонок приходит как `no_speech` или `silence_timeout` — как будто абонент молчал.	нет
`error`	`not_finalized`	Исход так и не был записан; `duration` равен `null`.	нет

Какой ответ оператора даёт какую причину

У причин корзины `not_connected` источник — код ответа оператора связи на наш вызов. Точность зависит от того, что публикует сервер оператора: когда кода нет, причиной остаётся родовой `dial_failed`, а сам код — в `error_code` рядом с ней.

Причина	Коды SIP
`no_answer`	408, 480, 487
`busy`	486, 600
`user_declined`	603
`invalid_destination`	404, 410, 484, 604
`spam_blocked`	607, 608
`provider_unavailable`	488, 493, 500, 502, 503, 504

Причина	Коды SIP
`trunk_auth_failed`	401; 403 — на транке с регистрацией
`sip_routing_error`	380, 400, 483
`dial_failed`	все остальные, в том числе 403 на транке peer-to-peer

403 на транке peer-to-peer (по IP) — недозвон, а не сбой транка. На таком транке логина нет, и 403 значит, что оператор не пускает звонок в это направление. Звонок получает `dial_failed`, код 403 — в `error_code`. На транке с регистрацией и если транк не найден 403 остаётся `trunk_auth_failed`.

`606 Not Acceptable` намеренно оставлен в `dial_failed`: один и тот же код приходит и от нехватки ёмкости на стороне оператора, и от отбоя автоответчика. Честной единой причины у него нет, поэтому разбирайте такие звонки по `error_code`.

Как выбирается причина

Правила применяются сверху вниз, побеждает первое сработавшее:

1. **Звонок не соединился** — причина корзины `not_connected` по ответу оператора; для входящего, где звонящий ушёл до ответа агента, — `caller_abandoned`.
2. **Исход определила платформа** — автоответчик, фильтр звонков или голосовое меню, перевод на оператора, тишина, предел длительности, инцидент наблюдателя или остановка на обновление.
3. **В разговоре нет ни одной распознанной реплики абонента** — `no_speech`; реплик или секунд меньше настроенного порога — `too_short`.
4. **Иначе решает, кто положил трубку** — сценарий дошёл до конца (`agent_hangup`) или абонент повесил трубку раньше (`user_hangup`).

> **`busy` и `user_declined` — разные причины.** `busy` означает «занято», `user_declined` — реальное отклонение вызова абонентом. Если вы строите отчёты по причинам недозвона, разводите их по этим двум значениям, а не по одному.
>
> Второе следствие для вашей логики повторов: **постоянна ли причина, зависит от рынка**. `user_declined` закрывает номер окончательно только на профиле США; на других профилях причина временная, и номер возвращается в набор, потратив попытку. Проверка запуска кампании отклоняет правило `expert`, которое повторяет постоянный исход (`rule_permanent_retry`).

Что кампания делает с номером после звонка

Так решает стандартная логика кампании. В экспертной логике так же обрабатывается исход, для которого нет правила или правило исчерпало `max_matches`; явное правило — `stop_calling`, `retry_in` или `wait_for_pass_end` — решает за неё.

Что произошло	`status_reason`	Что с номером	Попытка тратится
Не дозвонились	`no_answer`, `busy`, `dial_failed`	перезвон через интервал кампании	да
Ответил не человек	`voicemail_reached`, `voicemail_suppressed`, `screener_timeout`, `ivr_gate`	перезвон через интервал кампании	да
Абонент или его фильтр звонков отказал	`user_declined`, `screener_declined`	на профиле США — номер закрыт; на остальных — перезвон через интервал	да
Номер не годится	`invalid_destination`, `spam_blocked`	номер закрыт	да

Что произошло	status_reason	Что с номером	Попытка тратится
Сбой платформы на соединённом звонке	`internal_error`, `agent_unavailable`, `no_media`, `not_finalized`	перезвон через интервал кампании	да
Сбой телефонии до абонента	`trunk_auth_failed`, `provider_unavailable`, `sip_routing_error`	по умолчанию перезвон с нарастающей паузой: 1, 2, 4, 8 минут; после пятого такого сбоя номер «Исчерпан»	нет
Разговор состоялся	`agent_hangup`, `user_hangup`, `no_speech`, `too_short`, `silence_timeout`, `max_duration_reached`, `transferred_cold`, `transferred_warm`, `observer_incident`, `platform_shutdown`	номер закрыт	да
Звонок отменили до набора	`cancelled_by_function`, `cancelled_by_operator`	номер закрыт	нет

- **Перезвон назначается, только пока есть наборы и не закончилась кампания.** Если `attempts` израсходован или следующий набор пришёлся бы позже `ends_at`, номер становится «Исчерпан».
- **Интервал** — интервал кампании или ступень лестницы `intervals_s` для этого повтора (Поля кампании). Окно обзвона, реестр «не звонить» и частотные лимиты действуют на каждый перезвон.
- **Автоответчик перезванивается независимо от того, оставлено ли сообщение.** Сколько сообщений оставлять, решает политика голосовых сообщений, а не эта таблица.
- **Неизвестная причина закрывает номер:** если в оси появится новое значение, кампания не будет перезванивать по причине, которую не понимает.

Что путают чаще всего

`silence_timeout` и `no_speech` — одна ситуация с разным концом. Абонент молчит в обоих случаях; разница в том, кто положил трубку. Платформа — `silence_timeout`, абонент раньше неё — `no_speech`. Если вы считаете пустые звонки, складывайте обе причины.

Абонент молча положил трубку — это `no_speech`, а не `user_hangup`. Вторая причина ставится только там, где разговор был. Звонящий на входящий, положивший трубку до ответа агента, — `caller_abandoned`.

Нераспознанная речь считается молчанием. Шум, слишком тихая речь и обрыв без единого распознанного слова дают те же причины, что и тишина.

`silence_timeout` анализируется, `no_speech` — нет. Разница не в капризе настройки: абонент мог поговорить и замолчать в середине, поэтому тишина не означает пустой звонок.

Порог тишины отсчитывается от последней реплики агента, включая напоминания. Каждое напоминание вроде «Вы на линии?» запускает отсчёт заново. При пороге 12 секунд и двух напоминаниях через 8 секунд звонок с молчащим абонентом длится около 30 секунд, а не 12. Когда напоминания закончились, звонок завершается через интервал напоминаний после последнего — даже если порог тишины больше.

Правило повтора на автоответчик ловит оба случая. Условие по `voicemail_reached` срабатывает и тогда, когда причиной стало `voicemail_suppressed`.

Три причины фильтра звонков — это `completed`, а не `not_connected`. На звонок ответили — ассистент абонента или меню, — но человек не говорил. Если фильтр соединил с человеком, звонок получает обычную причину по итогу разговора; если фильтр предложил оставить сообщение и агент его оставил — `voicemail_reached`.

`health_reasons` — коды, объясняющие вердикт `warning` / `error`. Вердикт звонка — худший из сработавших: любой код уровня «ошибка» делает звонок `error`, иначе `warning`. Словарь открытый, новые коды со временем добавляются — незнакомый код не считайте ошибкой разбора.

Коды приходят из четырёх источников. Разделение важно: **в `health_reasons` попадает результат правила, а не его вход** — значения `status_reason` и внутренние коды событий здесь не встречаются никогда.

1. События звонка — что-то пошло не так внутри разговора:

Код	Уровень	Когда
`llm_execution_failed`	ошибка	исполняющая модель не ответила, агент произнёс запасную реплику
`llm_role_failed`	предупреждение	не ответила модель вспомогательной роли
`llm_retry`	предупреждение	ответ получен с запасной модели
`context_overflow`	ошибка	контекст модели переполнен
`tts_unspoken`	ошибка	синтез не состоялся, реплика не прозвучала
`tts_error`	предупреждение	сбой синтеза, от которого удалось оправиться
`tts_fallback`	предупреждение	переключение на запасной голос
`asr_turn_lost`	ошибка	реплика абонента потеряна распознаванием
`asr_error`	предупреждение	сбой распознавания без потери реплики
`realtime_error`	ошибка	сбой в realtime-режиме (speech-to-speech)
`noisy_line`	предупреждение	линия шумная: звонок шёл, распознавание не работало
`transfer_failed`	ошибка	перевод сорвался. Абонент, положивший трубку во время анонса перевода, — не сбой: у перевода исход `caller_left`, отметки нет
`transfer_ring_timeout`	предупреждение	назначение не ответило, отработала фолбэк-ветка
`observer_error`	предупреждение	сбой наблюдателя соответствия
`normalizer_degraded`	предупреждение	пре-TTS нормализатор не ответил
`lang_detect_degraded`	предупреждение	определение языка не ответило
`rag_degraded`	предупреждение	сбой обращения к базе знаний
`token_budget_degraded`	предупреждение	контекст урезан по токенному бюджету
`tool_failed`	предупреждение	функция или код-узел вернули ошибку
`silence_timeout`	предупреждение	автозавершение по молчанию абонента
`max_duration`	предупреждение	автозавершение по максимальной длительности

2. Ось статуса звонка — сюда сворачиваются значения `status_reason`:

Код	Уровень	Из каких `status_reason`
`disconnect_error`	ошибка	`internal_error`, `not_finalized`, `agent_unavailable`, `no_media`
`dial_infra_failed`	ошибка	`trunk_auth_failed`, `provider_unavailable`, `sip_routing_error`
`dial_failed`	предупреждение	остальные причины корзины `not_connected`

Причины на стороне абонента (`no_answer`, `busy`, `user_declined`, `invalid_destination`, `spam_blocked`) отмены по вашему решению до набора (`cancelled_by_function`, `cancelled_by_operator`) и входящие, брошенные до ответа (`caller_abandoned`), в вердикт **не входят**: это не дефекты платформы.

3. Пост-аналитика — `analysis_skipped`, `analysis_failed`, `analysis_interrupted`, все уровня «предупреждение», по терминальному состоянию разбора.

4. Запись и тайминги — `recording_head_lost` (часть записи не существует) и `slow_turns` (90-й процентиль длительности тёрна выше настроенного порога), оба «предупреждение».

ТЕХНИЧЕСКИЙ СПРАВОЧНИК

Вебхуки

События `call_ended` и `call_analyzed`, подпись, повторы и порядок доставки.

Вебхуки — основной канал результатов; опрос `GET /v1/calls/{call_id}` — запасной. Администратор регистрирует адрес **на агенте**: откройте агента в редакторе → вкладка **«Настройки»** → карточка **«Качество и интеграции»** → раздел **«Вебхуки»**; форма по полям описана на странице настроек «Вебхуки». Примеры полезной нагрузки — на странице «Вебхуки».

- **URL** — `http://` или `https://` (платформа не требует HTTPS; используйте его всё равно). Доставка подчиняется политике исходящих адресов платформы: приватные адреса и запрещённые хосты могут быть отклонены конфигурацией оператора, и такие доставки уходят в очередь недоставленных.
- **Подписка на события** — `call_ended`, `call_analyzed` или оба (по умолчанию оба).
- **Секрет подписи** — **ссылка на серверный секрет**: в форме администратор вписывает только имя переменной окружения развёртывания платформы. Само значение задаётся на стороне платформы, в кабинет не вводится и передаётся вам отдельным каналом.
- **Агент** — адрес привязан ровно к одному живому агенту организации. Поле обязательное: пустое имя агента отклоняется с кодом `webhooks.agent_required`, неизвестный или архивный — с кодом `webhooks.agent_not_found`. Уровня «вся организация» не бывает: чтобы покрыть несколько агентов, зарегистрируйте адрес у каждого (URL можно переиспользовать, различать по `agent_id` в payload).
- **Тестовые события** — отдельный флаг, при котором адрес получает и события тестовых сессий редактора (`mode: "test"`).
- **Таймаут** — сколько платформа ждёт ваш `2xx` (по умолчанию платформы 10 с, настраивается по адресу до 120 с; `0` означает значение платформы).

Конверт события

Каждое событие — HTTP(S) `POST` с телом JSON (UTF-8, не-ASCII символы не экранируются).

Поле	Описание
`event`	`call_ended` | `call_analyzed` (плюс `ping` для тестовых доставок)
`event_id`	Уникальный идентификатор события (`evt-` + 32 hex). Одинаков при повторных доставках и на разных адресах — **дедуплицируйте по нему**.
`timestamp`	Когда событие собрано, секунды эпохи.
`call_id`	Сопоставляется с объектом звонка.
`agent_id`	Имя агента.
`flow_version`, `flow_version_id`	Отметка версии сценария (та же оговорка о типе, что в объекте звонка) и идентификатор опубликованной версии.
`metadata`	Ваша нагрузка из создания звонка — присутствует в **каждом** событии. У звонка кампании её заполняет сама платформа, см. ниже.

Звонки кампании

Звонок, который сделала кампания, несёт в `metadata` кампанию и контакт, ради которого он набран. Чтобы узнать название кампании и где её создали, отдельный `GET /v1/campaigns/{campaign_id}` не нужен.

Ключ	Описание
`campaign_id`	Кампания.
`contact_id`	Контакт кампании. По нему контакт адресуют в операциях с контактами.

Ключ	Описание
`attempt_no`	Номер набора этого контакта, с 1. Повторные наборы после технического сбоя тоже считаются.
`campaign_name`	Название кампании в момент набора. Если кампанию потом переименовать, в уже сделанных звонках название не изменится.
`campaign_created_via`	Где создана кампания: `api` — через Web API, `web` — в кабинете, `legacy` — через адаптер совместимости, `""` — не записано (старые кампании).
`campaign_launched_via`	Где кампания запущена — те же значения.

Все ключи — строки, кроме `attempt_no`: это число. У звонка не из кампании этих ключей нет.

```
"metadata": {
  "campaign_id": "cam-b9b536a86e",
  "contact_id": "cc-41d7e0a2b9",
  "attempt_no": 2,
  "campaign_name": "Напоминания октябрь",
  "campaign_created_via": "api",
  "campaign_launched_via": "web"
}
```

Заголовки каждой доставки:

```
Content-Type: application/json
User-Agent: conversation-flow-webhooks/1.0
X-CFlow-Event: call_ended
X-CFlow-Event-Id: evt-4f0c22b17a9d4e3c8b6a5f019e2d7c31
X-CFlow-Timestamp: 1753344187
X-CFlow-Signature: v1=6f2a45c1e8…
```

Какое событие когда приходит

- **`call_ended`** — как только завершилась сессия, до анализа и обычно до финализации записи. Это раннее уведомление: транскрипт, `call_status`, `status_reason`, `error_code` и поля времени (`answered_at`, `media_started_at`, `talk_from`, `talk_to`). `latency` присутствует только если собирались метрики по тёрнам; её ключи — сырые имена метрик, оканчивающиеся на `_ms` (`turn_ms`, `llm_ms`, `asr_ms`, `tts_ms`, …), значения — `{avg, max}` в миллисекундах.
- **`call_analyzed`** — когда закончились **оба** пост-конвейера: анализ (терминальный в любом из `done`, `skipped`, `failed`, `interrupted` — событие приходит всегда) и финализация записи. Событие **самодостаточно**: полный блок звонка + анализ + готовая запись. Если нужен только итог, подпишитесь на один `call_analyzed` и пропустите `call_ended`. Его блок `call` идентичен `call_ended.call`, за исключением того, что запись остаётся на верхнем уровне и внутри `call` не дублируется.
- **`ping`** — тестовая доставка по кнопке в кабинете; `agent_id` — агент, к которому привязан адрес. Ответьте `2xx` и проверьте подпись как у настоящего события. Не повторяется.

Когда `analysis_status` не `done`, `call_analysis` равен `{}`, а поле верхнего уровня `reason` объясняет причину (например `voicemail`, `no_speech`, `analytics_disabled`); `analytics_tier`, `analytics_model_used` и `analytics_language` тогда пустые строки.

Семантика доставки

- **Не менее одного раза.** Доставка считается успешной только по `2xx` от вашего адреса; всё остальное (не-`2xx`, таймаут, ошибка соединения) повторяется. Дубли вы **будете** получать — дедуплицируйте по `event_id`.
- **Повторы с экспоненциальной задержкой** — это настройки платформы, а не адреса: по умолчанию до 8 попыток с паузами 30, 60, 120, 240, 480, 960 и 1920 секунд. После последней неудачной попытки событие уходит в очередь недоставленных, откуда администратор может доставить его вручную (История → детали

сессии → доставки вебхуков). Недоставленные со временем вычищаются, поэтому окно ручного повтора конечно.

- **Доставки всегда идут на текущий адрес**, поэтому опечатку в URL можно исправить и событие доставить заново; удалённый или отключённый адрес отправляет свои ожидающие события прямо в очередь недоставленных.
- **Порядок — по возможности.** `call_ended` попадает в очередь раньше `call_analyzed`, но порядок доставки не гарантирован. Обрабатывайте каждое событие независимо, сопоставляя по `call_id`.
- **Отвечайте быстро.** Возвращайте `2xx` сразу после проверки подписи и обрабатывайте асинхронно: платформа ждёт не дольше таймаута адреса.

Проверка подписи

```
signed_payload = "{X-CFlow-Timestamp}." + сырое тело запроса
X-CFlow-Signature: v1=<hex( HMAC-SHA256(secret, signed_payload) )>
```

`X-CFlow-Timestamp` — целое число секунд эпохи. Проверяйте по **сырым байтам** тела (до разбора JSON и повторной сериализации — тело в UTF-8 с неэкранированными не-ASCII символами), сравнивайте за постоянное время и отклоняйте устаревшие метки — окно 300 секунд. Повторные доставки переподписываются свежей меткой, поэтому окно с повторами не конфликтует.

Python

```python
import hashlib
import hmac
import time

def verify_webhook(
    secret: str,
    signature_header: str,   # X-CFlow-Signature, например "v1=6f2a45…"
    timestamp_header: str,   # X-CFlow-Timestamp, например "1753344187"
    raw_body: bytes,         # точные байты тела запроса
    tolerance_seconds: int = 300,
) -> bool:
    try:
        ts = int(timestamp_header)
    except (TypeError, ValueError):
        return False
    if abs(time.time() - ts) > tolerance_seconds:
        return False  # защита от переигровки
    signed = f"{ts}.".encode("utf-8") + raw_body
    expected = "v1=" + hmac.new(secret.encode("utf-8"), signed, hashlib.sha256).hexdigest()
    return hmac.compare_digest(expected, signature_header or "")
```

Node.js

```javascript
const crypto = require("crypto");

function verifyWebhook(secret, signatureHeader, timestampHeader, rawBody, toleranceSeconds = 300) {
  const ts = parseInt(timestampHeader, 10);
  if (!Number.isFinite(ts)) return false;
  if (Math.abs(Date.now() / 1000 - ts) > toleranceSeconds) return false;
  const expected =
    "v1=" +
    crypto
      .createHmac("sha256", secret)
      .update(`${ts}.`)
      .update(rawBody) // Buffer с точными байтами запроса
      .digest("hex");
  const provided = Buffer.from(signatureHeader || "");
  const wanted = Buffer.from(expected);
```

```
  return provided.length === wanted.length && crypto.timingSafeEqual(provided, wanted);
}
```

В Python (Flask) используйте `request.get_data()`; в Node (Express) — `express.raw({ type: "application/json" })`: оба сохраняют точные байты, по которым считалась подпись.

Записи разговоров

`recording_url` — **устойчивый** адрес (`https://{{HUBTALK_FQDN}}/api/recordings/rec-…`): он не истекает и его можно сохранять у себя. Каждый `GET` по нему отвечает `307` со свежей короткоживущей ссылкой на скачивание, поэтому адрес, сохранённый месяцы назад, продолжает работать — пока сама запись не удалена политикой хранения (`recording.expires_at`). Другие ответы: `409`, пока запись ещё не готова (продолжайте опрашивать), `410` после удаления, `503` при временной недоступности хранилища. Добавьте `?download=1`, чтобы отдавалось как вложение.

Для адреса нужен API-ключ организации, которой принадлежит запись, в том же заголовке, что у `/v1`: `Authorization: Bearer <API-ключ>`. Без ключа ответ — `401` `api_key_required`; неизвестный или отозванный ключ получает `401` `api_key_unknown` / `api_key_revoked`, действует лимит запросов `/v1` (`429`). Запись другой организации отвечает `404`, как несуществующая. Браузер, открывший адрес без ключа, попадает в кабинет: после входа открывается карточка звонка.

Особый случай: если запись финализируется необычно долго (дольше ожидания платформы, по умолчанию 30 с), `call_analyzed` всё равно отправляется — с текущим нетерминальным статусом записи и пустым `recording_url`. Устойчивый адрес начнёт работать сам, когда запись будет готова.

ТЕХНИЧЕСКИЙ СПРАВОЧНИК

Коды ошибок и пример от начала до конца

Полный перечень кодов и рабочий сценарий одним куском.

Коды ошибок

По типам

Статус	error.type	Коды на эндпоинтах этого справочника
400	`invalid_request_error`	`caller_number_not_allowed`, `caller_number_outside_grant_pool`, `destination_number_invalid`, `ring_duration_invalid`, `invalid_filter_value`, `invalid_cursor`
401	`authentication_error`	`api_key_required`, `api_key_unknown`, `api_key_revoked`
403	`permission_error`	`agent_not_in_subgroup`, `destination_not_allowed`, `privileged_key_required`, `root_org_key_required`
404	`not_found_error`	`trunk_not_found`, `subgroup_not_found`, `call_not_found`, `campaign_not_found`, `import_not_found`, `agent_not_found` (только веб-звонки)
409	`conflict_error`	`agent_subgroup_ambiguous`, `agent_not_published`, `dialing_stopped`, `launch_blocked`, `campaign_transition_invalid`, `campaign_closed`, `market_package_not_installed`, `idempotency_key_in_flight`, `dnc_decision_without_upload`, `dnc_screen_required`
422	`invalid_request_error`	`invalid_request` (схема) и коды кампаний ниже
429	`rate_limit_error`	`api_key_rate_limit_exceeded`, `dial_queue_full`, `subgroup_concurrency_limit`, `subgroup_rate_limit`
500	`api_error`	`internal_error`
502	`api_error`	`dial_failed` (только синхронный запасной режим), `contact_file_storage_unavailable`
503	`service_unavailable_error`	`voice_backend_unavailable`, `agent_dispatch_failed` (веб-звонки), `db_pool_exhausted` (платформа кратковременно перегружена; повторите через интервал из `Retry-After`, 1 с)

Коды ошибок кампаний

Статус	error.code	Причина
404	`campaign_not_found` / `import_not_found`	Такого объекта **в вашей организации** нет; всё, что за её границей, неотличимо от отсутствующего.
409	`launch_blocked`	Проверки запуска или возобновления не пройдены; список — в `error.blockers`.
409	`campaign_transition_invalid`	Переход невозможен из текущего состояния.
409	`campaign_closed`	`calls`, `contacts/withdraw`, `PATCH` скорости: кампания завершена или отменена.
409	`market_package_not_installed`	Контакты нельзя проверить для этой организации; обратитесь в поддержку.
409	`idempotency_key_in_flight`	Запрос с этим ключом ещё обрабатывается; дождитесь ответа.
409	`dnc_decision_without_upload`	Решение `dnc` у кампании, у которой нет загрузки.
409	`dnc_screen_required`	У загрузки нет построчной проекции: при решении `dnc`, при `?only=dnc_match` или как блокер запуска. Загрузите заново.

Статус	error.code	Причина
422	`invalid_request`	Тело не проходит проверку схемы; `message` — массив замечаний по полям.
422	`idempotency_key_required`	Заголовок `Idempotency-Key` отсутствует или пуст.
422	`idempotency_key_reuse`	Этот ключ уже использован для **другого** тела запроса.
422	`campaign_agent_unknown`	Агента с таким именем в вашей организации нет.
422	`campaign_window_invalid`	`ends_at` не позже `starts_at`.
422	`campaign_timezone_unknown`	Не название зоны IANA.
422	`campaign_classification_invalid`	`classification` не одно из трёх документированных значений.
422	`campaign_spec_invalid`	Отклонены длина названия, скорость вне 1–200 (при создании и в `PATCH`), настройка повторов, форма рабочих часов или действие по дублям.
422	`campaign_open_ends_at_required` / `campaign_open_ends_at_too_far`	У кампании, открытой к добавлению звонков, нет `ends_at` или он за горизонтом.
422	`campaign_auto_speed_ends_at_required` / `campaign_auto_speed_working_hours_required`	У кампании с `auto_speed: true` нет `ends_at` или, если она идёт несколько дней, нет `working_hours`. При создании и в `PATCH`.
422	`dialing_route_unresolved` / `dialing_route_ambiguous`	Нет привязки или больше одного кандидата.
422	`dialing_binding_incomplete`	`route_id` или `from_number` названы без `trunk_id`.
422	`working_hours_wider_than_law`	Расписание шире базового окна (законного или окна обзвона организации).
422	`attestation_import_id_required`	В `launch` нет `attestation.import_id`.
422	`dnc_decision_invalid`	`dnc` не `block` и не `skip`.
422	`contact_rows_rejected`	Запланированные звонки не удалось сохранить вовсе; читайте отчёт по загрузке.
422	`report_format_invalid`	`format` не `csv` и не `xlsx`.
422	`report_filter_invalid`	`only` не опущен и не `dnc_match`.
422	`calls_required` / `calls_batch_too_large`	`calls` пуст или содержит больше строк, чем предел (по умолчанию 1000).
422	`attestation_invalid`	`calls`: `attestation` не равна в точности `{"confirmed": true}`.
422	`calls_rejected`	Партия отклонена целиком из-за ошибки данных (или структурного отказа разбора); строки названы в `error.import`.
422	`calls_dnc_matched`	Партия отклонена целиком, потому что хотя бы один номер есть в реестре «не звонить»; читайте отчёт `dnc_match`.
502	`contact_file_storage_unavailable`	Источник контактов не удалось сохранить, загрузка не принята — повторите.

Пример от начала до конца

```
BASE_URL="https://app.hubtalk.kz"   # адрес вашего кабинета; Узбекистан — app.hubtalk.uz
API_KEY="cfk_…"   # секрет, показанный один раз при создании ключа
```

```
# 1. Посмотреть доступные агенту линии (необязательно: не присылайте
#    привязку вовсе, и платформа разрешит её сама)
curl -s "$BASE_URL/v1/dialing-routes?agent_id=collections-agent" \
  -H "Authorization: Bearer $API_KEY"

# 2. Создать кампанию, передать номера и запустить — один запрос
curl -s -X POST "$BASE_URL/v1/campaigns" \
  -H "Authorization: Bearer $API_KEY" \
  -H "Content-Type: application/json" \
  -H "Idempotency-Key: june-collections-2026-09-09" \
  -d '{
        "name": "june collections",
        "agent_id": "collections-agent",
        "calls_per_minute": 10,
        "timezone": "Asia/Almaty",
        "classification": "debt_collection",
        "working_hours": [{"days": [0,1,2,3,4], "start": "09:00", "end": "19:00"}],
        "attestation": {"confirmed": true},
        "planned_calls": [
          {"phone": "+77001234567", "name": "Aigerim", "variables": {"debt": "15000"}},
          {"phone": "+77007654321", "name": "Daniyar"}
        ]
      }'
# → 201. Смотрим "status": "running" | "scheduled" | "draft".
#   "draft" означает, что причина названа в "blockers": исправить и вызвать
#   POST /v1/campaigns/{id}/launch с
#   {"attestation": {"confirmed": true, "import_id": "<import.import_id>"}}
#   Учтите: по умолчанию retry_config.attempts = 1 — никого не перезванивают.

# 3. Дослать номера позже (кампания не должна быть закрыта)
curl -s -X POST "$BASE_URL/v1/campaigns/cam-b9b536a86e/calls" \
  -H "Authorization: Bearer $API_KEY" \
  -H "Content-Type: application/json" \
  -H "Idempotency-Key: crm-batch-2026-09-09-17" \
  -d '{"calls": [{"phone": "+77009998877", "name": "Symbat"}],
       "attestation": {"confirmed": true}}'
# → 201 (принята целиком) либо 422 calls_rejected / calls_dnc_matched (отклонена целиком)

# 4. Следить за ходом
curl -s "$BASE_URL/v1/campaigns/cam-b9b536a86e/progress" \
  -H "Authorization: Bearer $API_KEY"

# 5. Разобрать проблемные строки
curl -s "$BASE_URL/v1/imports/imp-353d158dd2/report?format=csv" \
  -H "Authorization: Bearer $API_KEY" -o import-report.csv

# ...и отдельно совпадения с реестром «не звонить»
curl -s "$BASE_URL/v1/imports/imp-353d158dd2/report?only=dnc_match&format=csv" \
  -H "Authorization: Bearer $API_KEY" -o import-dnc.csv

# 6. Результаты по каждому звонку придут вебхуками call_ended / call_analyzed —
#    кампания набирает те же самые звонки, что этот API размещает по одному
```