

SPEAKNODE

Описание функциональных характеристик

Версия 1.0



Настоящая документация может быть использована только для поддержки работоспособности продуктов, поставленных компанией Voxagent.

Все примеры конфигураций, реквизитов и других данных в документе вымышлены и не относятся к реальным сущностям. Любое совпадение случайно.

Все встречающиеся в тексте товарные знаки и зарегистрированные товарные знаки являются собственностью их владельцев и используются исключительно для идентификации программного обеспечения или компаний.

Данная документация может не отражать некоторых модификаций программного обеспечения. Если вы заметили ошибки или опечатки, сообщите об этом по контактными данным ниже.

Все имущественные авторские права сохраняются за Voxagent в соответствии с действующим законодательством.

© Voxagent, 2026

Россия

help@mail.voxagent.app · voxagent.app

Содержание

Обзор функционала
Основные возможности
Как это работает
AI-агенты
Настройка агента
Основные параметры
Языки и локализация
LLM-модель
Распознавание речи (STT)
Детекция голосовой активности (VAD)
Голос (TTS)
Динамические переменные
Инструменты агента
Публикация
Дублирование агента
Аналитика
История разговоров
Список сессий
Детали сессии
Телеметрия и трейсинг
Пайплайн телеметрии
Хранение данных
Биллинг
Модель биллинга
Подписки и тарифы
Отслеживание использования
Обработка платежей
Управление биллингом
Исходящие кампании
Обзор
Настройка кампании
Элементы кампании

Как это работает
Мониторинг
Команды и пространства
Пространства
Что принадлежит пространству
Управление пространством
Роли пользователей
Приглашения
Аутентификация
Телефония
Поддерживаемые провайдеры
Входящие звонки
Управление номерами
Исходящие звонки
Интеграция с VoxImplant
SIP Trunk (LiveKit)
Инструменты и интеграции
Обзор
Типы инструментов
Webhook-инструменты
Системные инструменты
Настройка инструмента
Речь во время выполнения инструмента
Правила переключения
Как это работает
Управление вебхуками
Примеры
Голос и коммуникация
Как это работает
Голосовой пайплайн
Распознавание речи (STT)
Языковая модель (LLM)
Синтез речи (TTS)
Детекция голосовой активности (VAD)
Запись аудио

- [Уведомления в реальном времени](#)
- [Жизненный цикл сессии](#)
- [Виджет для сайта](#)
- [Обзор](#)
- [Установка](#)
- [Настройка](#)
- [Атрибуты](#)
 - [Передача переменных через URL](#)
- [Настройка виджета в админ-панели](#)
- [Как это работает](#)
- [Требования](#)

Описание функциональных характеристик

Обзор функционала

Speaknode — платформа для создания и управления голосовыми AI-агентами, которые общаются с пользователями по телефону, через виджет на сайте или напрямую в браузере.

Основные возможности

Функция	Описание
AI-агенты	Создание и настройка голосовых агентов с кастомным поведением, личностью и знаниями
Голос и коммуникация	Голосовые разговоры в реальном времени на базе WebRTC
Виджет для сайта	Встраиваемый виджет для добавления голосового агента на любой сайт
Телефония	Входящие и исходящие звонки через Twilio и VoxImplant
Исходящие кампании	Массовые исходящие обзвоны с динамическими переменными
Инструменты и интеграции	Подключение агентов к внешним API и вебхукам
Аналитика	История разговоров, записи и телеметрия
Команды и пространства	Рабочие пространства для совместной работы
Биллинг	Тарификация по использованию

Как это работает

Пользователь настраивает агента → Агент запускается в LiveKit-комнате → AI-агент подключается и начинает разговор → Сессия записывается и анализируется

1. **Настройте** — задайте поведение агента: системный промпт, голос, язык, инструменты
2. **Разверните** — встройте виджет на сайт, подключите номер телефона или используйте API
3. **Общайтесь** — агент ведёт разговоры в реальном времени
4. **Анализируйте** — просматривайте записи, транскрипции и данные телеметрии

AI-агенты

AI-агенты — основа Speaknode. Каждый агент — это настраиваемый голосовой ассистент, способный вести разговоры с пользователями в реальном времени.

Настройка агента

Основные параметры

- **Название** — имя агента (до 50 символов)
- **Системный промпт** — инструкции, определяющие поведение, личность и знания агента. Поддерживает динамические переменные в формате `{{имя_переменной}}`
- **Temperature** — контролирует случайность ответов LLM (0 = детерминированные, 1 = креативные)
- **Лимит токенов** — максимальное потребление токенов за разговор

Языки и локализация

Каждый агент поддерживает несколько языков. Для каждого языка настраивается:

- **Первое сообщение** — приветствие, которое агент произносит в начале разговора
- **TTS-голос** — голос и провайдер синтеза речи
- **Скорость речи** — множитель скорости воспроизведения
- **Фоновый звук** — звуковой фон во время разговора (тишина, офис, кофейня и т.д.)

Агент может автоматически определять язык пользователя и переключаться на соответствующую локаль.

LLM-модель

Выбор языковой модели, которая обеспечивает логику агента:

- LLM-провайдер и версия модели

- Temperature и дополнительные параметры модели

Распознавание речи (STT)

Настройка транскрибации речи пользователя:

- Модель и STT-провайдер
- Параметры, специфичные для модели

Детекция голосовой активности (VAD)

Управление детекцией речи пользователя:

- Выбор модели VAD
- Параметры чувствительности и таймингов
- Возможность отключить прерывания во время первого сообщения

Голос (TTS)

Настройка голоса агента для каждого языка:

- TTS-провайдер
- Выбор голоса и его параметры
- Регулировка скорости
- Тип и громкость фонового звука

Динамические переменные

Агенты поддерживают динамические переменные в системных промптах и первых сообщениях. Переменные задаются через OpenAPI-схему и подставляются при запуске.

Пример системного промпта:

```
Ты агент поддержки компании {{company_name}}.  
Имя клиента – {{customer_name}}, номер заказа – {{order_id}}.
```

Переменные передаются через:

- API-вызов при диспатче
- URL-параметры (для виджета)
- Данные элемента кампании (для исходящих звонков)

Каждая переменная имеет:

- Имя
- Тип (string, number, boolean, integer)

- Описание
- Значение по умолчанию
- Флаг nullable

Инструменты агента

Агенты могут вызывать внешние API во время разговора. Подробнее в разделе [Инструменты и интеграции](#).

Публикация

Агент может быть опубликован для предоставления доступа:

- **Опубликованный агент** — доступен по публичной ссылке для тестирования
- **Виджет** — встраивается на любой сайт
- **Номер телефона** — привязывается к входящей/исходящей телефонии
- **API** — запускается программно

Дублирование агента

Любой агент можно продублировать, создав копию со всеми настройками. Полезно для создания вариаций или шаблонов.

Аналитика

Speaknode собирает детальные данные о каждом разговоре для анализа и мониторинга.

История разговоров

Каждая сессия агента записывается с:

- **Метаданные сессии** — агент, звонящий, время начала/окончания, длительность, статус
- **Аудиозапись** — полное аудио сессии в объектном хранилище
- **Транскрипция** — текстовая расшифровка разговора
- **Телеметрия** — детальные OpenTelemetry-трейсы каждого шага

Список сессий

Просмотр всех разговоров с фильтрами:

- По агенту
- По статусу (завершён, ошибка, активен)

- По диапазону дат
- По информации о звонящем

Детали сессии

Для каждой сессии можно просмотреть:

- Полную транскрипцию с разбивкой по репликам
- Воспроизведение аудио с визуализацией волны
- Метрики по каждой реплике (задержка, длительность)
- Вызовы инструментов во время сессии
- Детали ошибок (если произошёл сбой)

Телеметрия и трейсинг

Платформа использует OpenTelemetry для сквозной трассировки:

```
Пользователь говорит → STT span → LLM span → Tool call span → TTS span →  
Пользователь слышит
```

Каждый разговор — это трейс, содержащий спаны для каждой операции:

- **Имя спана** — тип операции (STT, LLM inference, tool call, TTS)
- **Длительность** — сколько времени занял каждый шаг
- **Статус** — успех или ошибка
- **Атрибуты** — теги и метаданные
- **События** — детальные события внутри спана

Пайплайн телеметрии

```
Python Worker → OTEL Collector → Kafka → ASP.NET Consumer → База данных  
→ Langfuse (внутренний мониторинг)
```

Хранение данных

Данные	Хранилище
Метаданные сессий	PostgreSQL
Аудиозаписи	S3-совместимое объектное хранилище
Спаны телеметрии	PostgreSQL (через Kafka)
Агрегации аналитики	ClickHouse
LLM-трейсы	Langfuse

Биллинг

Speaknode использует тарификацию по использованию на базе GetLago.

Модель биллинга

Тарификация основана на **голосовых минутах** — фактическом времени разговоров с AI-агентами.

Подписки и тарифы

Каждое пространство имеет тарифный план, определяющий:

- Включённые голосовые минуты за период
- Стоимость превышения
- Уровни доступа к функциям
- Лимиты по агентам

Отслеживание использования

Платформа автоматически отслеживает использование:

1. Каждая сессия агента фиксирует время начала и окончания
2. Длительность рассчитывается и передаётся в GetLago
3. GetLago агрегирует использование относительно тарифа
4. Счета формируются в конце биллингового периода

Обработка платежей

Платежи проходят через биллинговый движок платформы с подключённым защищённым платёжным провайдером:

- Автоматическая синхронизация платёжных методов
- Генерация и доставка счетов
- Биллинг за превышение

Управление биллингом

В панели управления:

- Просмотр текущего использования и оставшихся минут
- Управление тарифным планом
- Просмотр истории счетов
- Обновление платёжного метода

Исходящие кампании

Speaknode позволяет запускать массовые исходящие обзвоны с помощью AI-агентов.

Обзор

Кампания — это набор исходящих звонков по списку контактов. Каждый звонок обрабатывается AI-агентом с динамически подставленными переменными для каждого контакта.

Настройка кампании

1. **Выберите агента** — укажите AI-агента, который будет обрабатывать звонки
2. **Добавьте контакты** — загрузите список с номерами телефонов и данными переменных
3. **Настройте переменные** — сопоставьте данные контакта с переменными агента (например, `{{customer_name}}`, `{{order_id}}`)
4. **Запустите** — платформа последовательно обзванивает каждый контакт

Элементы кампании

Каждый контакт в кампании — это **элемент кампании** с:

- Номером телефона для звонка

- Значениями переменных для конкретного звонка
- Отслеживанием статуса (ожидание, в процессе, завершён, ошибка)
- Ссылкой на сессию разговора после звонка

Как это работает

1. Создаётся кампания со списком элементов
2. Для каждого элемента платформа:
 - Создает LiveKit-комнату
 - Запускает агента с переменными элемента
 - Иницирует исходящий звонок через телефонного провайдера
 - Отслеживает статус сессии
3. Результаты доступны по каждому элементу (запись, транскрипция, статус)

Мониторинг

Прогресс кампании отображается в панели управления:

- Общий статус кампании
- Результаты звонков по каждому элементу
- Ссылки на отдельные сессии разговоров

Команды и пространства

Speaknode поддерживает командную работу через рабочие пространства — **Spaces**.

Пространства

Space — это общее рабочее пространство, где участники команды совместно работают над агентами, кампаниями и настройками.

Что принадлежит пространству

- Агенты и их конфигурации
- Номера телефонов
- Инструменты и интеграции
- Кампании
- Вебхуки
- Биллинг и подписка

Управление пространством

- **Создание** нового пространства для команды
- **Приглашение** участников по email
- **Управление** ролями и правами участников

Роли пользователей

Роль	Возможности
Owner	Полный доступ, управление биллингом, управление участниками
Member	Создание и управление агентами, кампаниями, инструментами
Admin (платформа)	Настройки платформы, управление моделями, контроль пользователей

Приглашения

Участники приглашаются по email. Процесс приглашения:

1. Владелец пространства отправляет приглашение
2. Приглашённый получает email со ссылкой активации
3. Пользователь регистрируется или входит через Keycloak SSO
4. Пользователь добавляется в пространство

Аутентификация

Все пользователи авторизуются через Keycloak (OpenID Connect):

- Вход по email/паролю
- Интеграция с Google OAuth
- Поддержка SSO
- Управление сессиями на основе токенов

Телефония

Speaknode интегрируется с телефонными провайдерами, позволяя AI-агентам принимать и совершать телефонные звонки.

Поддерживаемые провайдеры

Провайдер	Входящие	Исходящие	Примечания
Twilio	Да	Да	Прямая интеграция через SIP trunk
VoxImplant	Да	Да	Через WebSocket Media Bridge (нет прямого доступа к SIP)

Входящие звонки

При поступлении звонка на подключённый номер:

1. Телефонный провайдер отправляет вебхук на платформу
2. Вебхук принимается через Kafka
3. Бэкенд определяет агента, привязанного к номеру
4. Создаётся LiveKit-комната с SIP-участником
5. AI-агент запускается в комнате
6. Разговор начинается

Управление номерами

- Импорт существующих номеров из Twilio или VoxImplant
- Привязка номера к конкретному агенту
- Настройка параметров для каждого номера

Исходящие звонки

Исходящие звонки можно инициировать:

- Индивидуально через API
- Массово через [Исходящие кампании](#)

Платформа создаёт LiveKit-комнату, запускает агента и инициирует исходящий звонок через настроенного телефонного провайдера.

Интеграция с VoxImplant

VoxImplant не предоставляет прямой доступ к SIP trunk. Платформа использует **WebSocket Media Bridge** — сервис на Node.js, который:

1. Получает аудио от VoxImplant через WebSocket
2. Стримит его в LiveKit-комнату

3. Отправляет аудио из LiveKit обратно в VoxImplant

Это обеспечивает полную функциональность агента с VoxImplant в качестве телефонного бэкенда.

SIP Trunk (LiveKit)

LiveKit Server включает встроенный SIP-компонент, который обеспечивает:

- Регистрацию SIP trunk
- Правила диспатча для маршрутизации звонков в комнаты
- Прямой мост SIP-to-WebRTC (для провайдеров с поддержкой SIP)

Инструменты и интеграции

Агенты Speaknode могут вызывать внешние API во время разговоров, получая данные, выполняя действия и интегрируясь с вашими системами.

Обзор

Инструменты — это HTTP-интеграции, которые LLM может вызывать во время разговора. Когда агент решает, что ему нужны внешние данные или нужно выполнить действие, он вызывает настроенный инструмент и использует результат в своём ответе.

Типы инструментов

Webhook-инструменты

Пользовательские HTTP-эндпоинты, которые агент может вызывать:

- **Метод** — GET, POST, PUT, DELETE
- **URL** — адрес эндпоинта (поддерживает подстановку переменных)
- **Заголовки** — кастомные заголовки, определённые через OpenAPI-схему
- **Тело** — тело запроса, определённое через OpenAPI-схему
- **Описание** — объясняет LLM, когда и как использовать инструмент

Системные инструменты

Встроенные инструменты платформы (например, определение языка, перевод разговора).

Настройка инструмента

Каждый инструмент определяется OpenAPI-схемой для его параметров:

- **Схема тела** — определяет структуру JSON-тела, которое LLM должен сформировать
- **Схема заголовков** — определяет кастомные заголовки для отправки
- **Код** — уникальный идентификатор, по которому LLM ссылается на инструмент

Речь во время выполнения инструмента

Инструменты поддерживают настраиваемую речь при выполнении:

- **Pre-tool speech** — что агент говорит перед вызовом инструмента (например, «Сейчас проверю...»)
- **Progress speech** — что агент говорит в ожидании ответа

Правила переключения

Инструменты могут включать правила переключения, перенаправляющие разговор:

- **Переключение на другого агента** — передача разговора другому AI-агенту
- **Условие** — когда должно произойти переключение

Это позволяет создавать мультиагентные сценарии, где специализированные агенты обрабатывают конкретные темы.

Как это работает

1. LLM решает вызвать инструмент на основе контекста разговора
2. Python worker формирует HTTP-запрос по OpenAPI-схеме
3. Запрос отправляется на настроенный эндпоинт
4. Ответ передаётся обратно в LLM
5. LLM включает результат в свой ответ

Управление вебхуками

Платформа также предоставляет отдельную систему вебхуков для приёма событий:

- Настройка эндпоинтов для получения событий сессий
- Просмотр истории запросов и полезных нагрузок
- Управление разрешёнными путями и маршрутизацией через Webhook Receiver Client

Примеры

Поиск в CRM:

```
Инструмент: get_customer_info
Метод: GET
URL: https://crm.example.com/api/customers/{{customer_id}}
Описание: Поиск информации о клиенте по ID
```

Статус заказа:

```
Инструмент: check_order_status
Метод: POST
URL: https://api.example.com/orders/status
Тело: { "order_id": "string" }
Описание: Проверить текущий статус заказа клиента
```

Голос и коммуникация

Speaknode использует WebRTC для голосовой связи в реальном времени между пользователями и AI-агентами.

Как это работает

1. При запуске агента создаётся **LiveKit-комната**
2. **AI-агент** (Python worker) подключается к комнате и начинает слушать
3. **Пользователь** подключается через браузер (виджет или клиент) или телефон
4. Аудио передаётся в обе стороны в реальном времени
5. Сессия **записывается** автоматически (audio egress)

Голосовой пайплайн

```
Пользователь говорит → WebRTC аудио → STT (речь в текст) →
LLM обрабатывает текст → генерирует ответ →
TTS (текст в речь) → WebRTC аудио → Пользователь слышит
```

Распознавание речи (STT)

Преобразует речь пользователя в текст для LLM. STT-провайдер настраивается для каждого агента отдельно.

Языковая модель (LLM)

Обрабатывает разговор и генерирует ответы. LLM-провайдер и модель настраиваются для каждого агента отдельно (напрямую или через агрегатор).

Синтез речи (TTS)

Преобразует ответы агента в речь. TTS-провайдер настраивается для каждого агента отдельно.

Детекция голосовой активности (VAD)

Определяет, когда пользователь начинает и прекращает говорить. Управляет поведением очередности — когда агент должен начать или прекратить говорить.

Запись аудио

Каждый разговор автоматически записывается через LiveKit Egress. Записи включают:

- Полное аудио сессии (композитная запись)
- Треки отдельных участников
- Хранение в S3-совместимом объектном хранилище

Уведомления в реальном времени

Платформа использует SignalR для уведомления фронтенда о событиях сессии:

- **Agent Ready** — агент подключился к комнате
- **Session Completed** — разговор завершён, аудио доступно

Жизненный цикл сессии

Статус	Описание
Pending	Сессия создана, комната ещё не готова
Dispatching	LiveKit-комната создана, ожидание подключения агента
Active	Агент подключился, разговор идёт
Completed	Разговор завершён нормально
Failed	Произошла ошибка во время сессии

Виджет для сайта

Speaknode предоставляет встраиваемый веб-компонент, который добавляет голосового агента на любой сайт.

Обзор

Виджет — это Angular веб-компонент, упакованный как `<speacknode-agent>`. Он подключается к опубликованному агенту и позволяет посетителям сайта вести голосовой разговор прямо в браузере.

Установка

Разместите кастомный элемент в нужном месте страницы, а скрипт-загрузчик добавьте либо сразу после элемента, либо перед `</body>`:

```
<speacknode-agent agent-id="your-agent-id"></speacknode-agent>
<script src="https://app-widjet.speaknode.com/speaknode-widget-loader.js">
</script>
```

Скрипт-загрузчик регистрирует кастомный элемент `<speacknode-agent>`, подгружает конфигурацию опубликованного агента и инициализирует виджет, когда пользователь нажимает кнопку начала разговора.

Настройка

Атрибуты

Атрибут	Тип	Описание
<code>agent-id</code>	string	ID опубликованного агента
<code>agent-token</code>	string	Предгенерированный JWT-токен (опционально)
<code>language</code>	string	Переопределение языковой локали
<code>initialized</code>	boolean	Режим внешней конфигурации

Передача переменных через URL

Динамические переменные можно передавать как query-параметры URL. Виджет считывает их автоматически и передаёт агенту при запуске.

Пример:

```
https://yoursite.com/page?customer_name=Иван&order_id=12345
```

Настройка виджета в админ-панели

На странице редактирования агента вкладка **Widget** позволяет:

- Настроить расположение и внешний вид виджета
- Сгенерировать код для встраивания
- Задать кастомные цвета темы

Как это работает

1. Виджет загружается на странице и получает публичную конфигурацию агента
2. Пользователь нажимает, чтобы начать разговор
3. Виджет запрашивает токен диспатча у бэкенда
4. Создаётся LiveKit-комната, запускается агент
5. Виджет подключается к комнате через WebRTC
6. Голосовой разговор начинается в реальном времени
7. После завершения разговора сессия сохраняется

Требования

- Агент должен быть **опубликован** (`is_published: true`)
- Браузер пользователя должен поддерживать WebRTC
- Необходим доступ к микрофону