

# 1С:Предприятие 8

## **Платформа интеграции 1С с чат-ботами**

Версия 5.xx

Инструкция (демо)

Г. Иннополис

Разработчики «Бот 1С»

2020

# ОГЛАВЛЕНИЕ

<b>Спецификация .....</b>	<b>3</b>
Назначение .....	3
Функциональность .....	3
Требование к программному окружению .....	3
Спецификация функциональных возможностей .....	4
<b>Руководство по настройке .....</b>	<b>5</b>
3.1. Создание и настройка чат-бота .....	5
3.2. Сценарии поведения чат-бота .....	7
3.3. Графический редактор сценариев (версия Enterprise) .....	8
<b>Руководство разработчика, API .....</b>	<b>15</b>
4.1. Подключаемые методы (внешняя обработка) .....	15
4.2. API .....	18

# СПЕЦИФИКАЦИЯ

## Назначение

Система интеграции чат-ботов с 1С предназначена для взаимодействия программы 1С с системами чат-ботов. Поддерживается работа с чат-ботами следующих систем: Telegram, Viber, Skype, WhatsApp, Facebook, Одноклассники, ВКонтакте, Яндекс.Алиса. Результатом является взаимодействие клиента через чат-боты мессенджеров вышеуказанных систем с системой 1С.

## Функциональность

- Взаимодействие с системами чат-ботов (прием и отправка сообщений, мультимедиа данных, отображение меню): Telegram, Viber, Skype, Facebook, Одноклассники, ВКонтакте, Яндекс.Алиса, WhatsApp. Взаимодействие с клиентом осуществляется только в private – режиме.
- Механизм автоматического подбора сценария и выполнения его действий.
- Визуальный конструктор сценариев, с динамическим применением изменений.
- Предоставление средств использования возможности системы по взаимодействию с чат-ботами для других модулей системы.
- Взаимодействие происходит напрямую с официальными открытыми и разрешенными API мессенджеров, не используя другие сервисы и не передавая данные 3-им лицам. Исключение составляет мессенджер WhatsApp, для работы которого задействован сторонний сервис <https://chat-api.com>.

## Требование к программному окружению

**Операционная система:** Не имеет значения

**Система языка:** 1С

**Платформа:**

1С:Предприятие 8.3.7 и выше без режима совместимости или с ним не менее указанной версии.

Режимом интерфейса: Такси (Управляемые формы)

**Конфигурация:** УТ - 11.2.2 и выше,

Розница - 2.1.8 и выше

УНФ - 1.5.4 и выше

CRM - 2.0 и выше

*Примечание: Так же возможна установка на любой конфигурации 1С: Совместимо с небольшими изменениями 3х запросов на объекты кастомной конфигурации.*

## Спецификация функциональных возможностей

	Telegram	Viber	Facebook	Skype	ВКонтакте	Одноклассники	Яндекс.Алиса	WhatsApp*
Проверка подключения	+	+	+	+	+	+	+	+
Чтение данных аккаунта	+	+	1*	1*	1*	+	1*	1*
Установка вэб-хука	+	+	2*	2*	2*	+	2*	+
Прием сообщений	+	+	+	+	+	+	+	+
Прием мультимедиа данных	+	+	+	+	+	+	+	+
Запрос номера телефона	+	+	3*	3*	3*	3*	3*	+
Запрос местоположения	+	+	4*	4*	4*	4*	4*	4*
Получение данных клиента	+	+	+	+	+	+	5*	+
Отправка сообщений	+	+	+	+	+	+	+	+
Отправка мультимедиа данных	+	+	+	+	+	+	+	+
Отправка контакта	+	+	6*	6*	6*	6*	6*	6*
Отправка местоположения	+	+	7*	7*	+	7*	7*	7*
Отправка клавиатуры меню	+	+	+	+	+	8*	+	8*
Отправка клавиатуры в тексте	+	9*	9*	9*	9*	9*	9*	9*

1\* - Не поддерживается внешней системой, при успешном подключении заполняется фиксированными значениями «\*\*\*\*\*».

2\* - Не поддерживается внешней системой, необходимо скопировать адрес из 1С и вручную вставить в личном кабинете аккаунта внешней системы.

3\* - Не поддерживается внешней системой, генерируется отправка сообщение с просьбой написать номер телефона, и после ответа отображается оповещение сотруднику на подтверждение.

4\* - Не поддерживается внешней системой, генерируется отправка сообщения с просьбой отправить свой адрес.

5\* - Не поддерживается внешней системой.

6\* - Не поддерживается внешней системой, имя контакта и номер телефона приводятся к тексту и отправляются сообщением.

7\* - Не поддерживается внешней системой, координаты приводятся и генерируется ссылка на карту google.map с указанием точки.

8\* - Не поддерживается внешней системой, клавиатура генерируется текстом и отправляется сообщением.

9\* - Не поддерживается внешней системой, приводится к клавиатуре меню.

WhatsApp\* - С данной системой взаимодействие происходит не напрямую а через поставщика API - <https://chat-api.com>, в связи с чем необходимо дополнительно обращаться к данной организации на предоставление услуг (стоимость 900 руб/мес). Поэтому данная система может не включаться в состав лицензии.

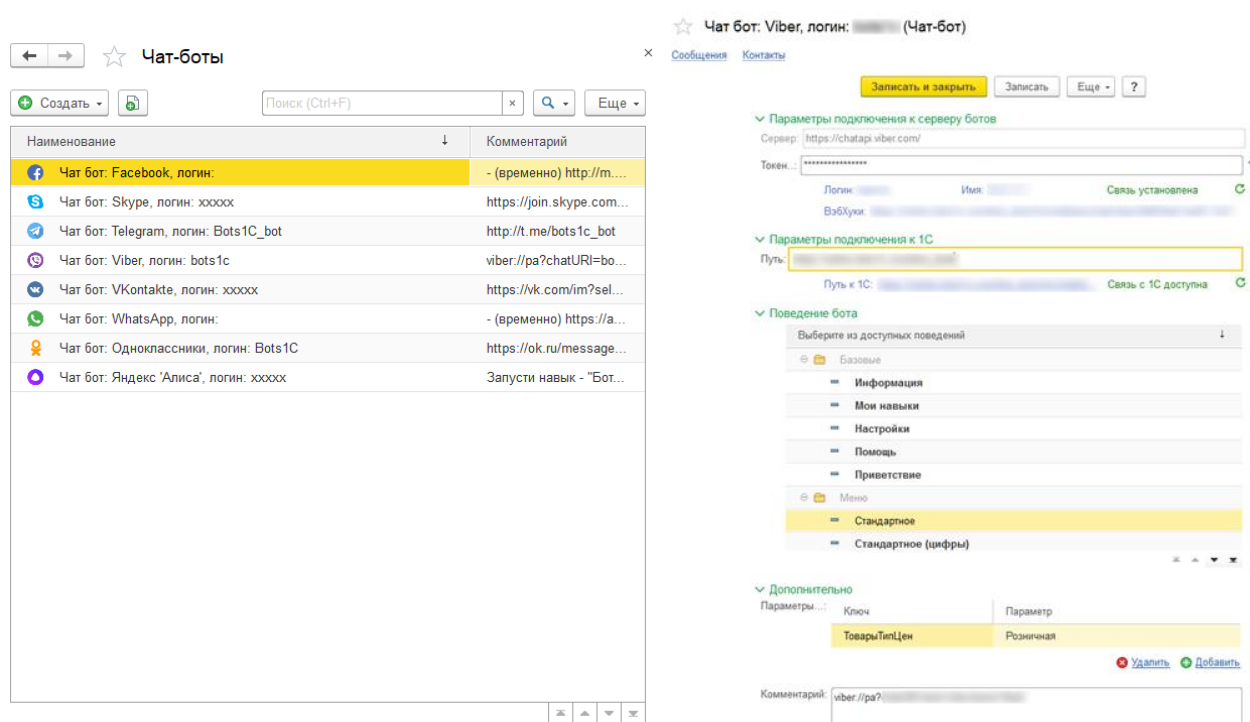
# РУКОВОДСТВО ПО НАСТРОЙКЕ

В данном разделе рассматривается добавление и настройка чат-бота, сценариев его работы, подключение дополнительных механизмов выполнения произвольного метода.

## 3.1. СОЗДАНИЕ И НАСТРОЙКА ЧАТ-БОТА

Чат-боты описываются в соответствующем справочнике «Чат-боты». В форме списка отображается представление (тип + логин) и комментарий.

Для добавления нового чат-бота, в форме списка необходимо нажать кнопку «создать» и выбрать подходящий тип системы. Далее в форме создания элемента заполняем поля:



### Параметры подключения к серверу ботов:

В данной группе содержатся поля для подключения к внешней системе аккаунта чат-бота.

- Сервер – данное поле недоступно и проставляется автоматически. Содержит URL путь к серверу соединения с системой мессенджера. Для WhatsApp оно вводится вручную по данным из внешней системы. Если у вас на сервере ограничение по открываемым URL, то вам необходимо данный путь добавить в исключения.
- Данные прокси (только для телеграмма): Появляются дополнительные 5-ть полей для указания данных прокси, если их заполнить, то связь с внешней системой чат-бота будет идти через прокси. Т.е. будет работать на территории РФ.
- Токен – это ключ, который вы получаете во внешней системе мессенджера после создания аккаунта чат-бота. Если он указан верно – то после перехода на следующий элемент появятся данные об аккаунте (Или, если система не предполагает чтение логина – то текст «xxx») и отобразится статус «Соединение установлено». Можно нажать

вручную кнопку «Тест соединения» - если удалось подключиться к системе – то статус поменяется на «успешно», в случае ошибки –установится статус с красным текстом и отобразится сообщения расшифровки ошибки.

- ID – Данное поле является дополнительным к токену, и появляется только на некоторых типах систем (Skype, Яндекс.Алиса). При создании аккаунтов чат-ботов ([Раздел 2: Руководство по созданию чат-ботов](#)), описано из каких полей его брать.

#### **ВНИМАНИЕ:**

- На территории РФ, для работы телеграмма, необходимо использовать прокси, его можно приобрести в том числе на сайте <https://proxуb.net>, (примерная стоимость 35 руб/мес.).

- Замечено что на территории РФ, в Москве и некоторых других регионах, прием сообщений от телеграм дополнительно глушат операторы МТС. Для проверки можно временно подключить vpn или переключиться на другого оператора.

- Если не подключается к внешней системе, то проблема может быть в: Интернет соединении с приложения 1С, недоступности/заблокированном соединении до URL, неверно указанного токена и/или ID приложения.

#### **Параметры подключения к 1С**

В этом разделе описываются поля, для подключения внешней системы мессенджера к 1С для отправки новых сообщений от клиента.

- Путь – указывается путь к публикации базы 1С, в формате: [https://online.site.ru/ut\\_bots](https://online.site.ru/ut_bots)  
Где: online.site.ru – Доменное имя сервера (Указывать IP не верно!)  
ut\_bots – Имя публикации
- Код – Для системы ВКонтакте необходимо добавить код подтверждения, он получается из системы ВК при создании бота.

#### **ВНИМАНИЕ:**

- В фейсбук, вконтакте, Яндекс.Алиса, Skype не поддерживают автоматическая установка вэб хуков. Для установки, необходимо сохранить чат-бота, и нажать на гиперссылку «Путь к 1С», далее в открывшемся браузере, с адресной строки, скопировать URL, и вставить его в соответствующее поле на странице аккаунта чат-бота (описано в [разделе 2: Руководство по созданию чат-ботов](#)).

- В пути к 1С обязательно указывать только доменное имя, если указать IP адрес то работать не будет.

- Если не проходит соединение, причины могут быть следующие:

Не работает публикация.

Не прописан или не верные данные пользователя для автоматической авторизации.

Не запущен или некорректно работает IIS, Apache.

Не верно установлен сертификат в IIS, Apache

Не сброшен порт (443) с роутера на сервер

Не проходит запрос с доменного имени на роутер

Описание тех или иных проверок описаны в разделе 1: Руководство по установке.

## Сценарии чат-бота

Основой взаимодействия Чат-бота является сценарий. Он определяет все шаги и сообщения чат-бота. Для любого чат-бота можно установить произвольный набор сценариев. При этом один и тот же сценарий может быть на различных системах чат-бота.

- **Меню старта** – Здесь устанавливается ссылка на сценарий, который, после завершения сценария, будет запущен. Как правило это главное меню для пользователя.
- **Не подобран** – Сценарий, который будет запущен если не был подобран ни один подходящий сценарий, по ключевым словам, которые определяются в условиях наступления.
- **Выбранные сценарии** – В данной таблице отображаются все сценарии. Чтобы подключить сценарий, достаточно щёлкнуть по нему, при этом он должен выделиться жирным шрифтом, если это произошло – значит сценарий активирован для работы с данным чат-ботом.

Рекомендуется для всех устанавливать одни и те же сценарии, исключение составляет Яндекс.Алиса, из-за особенностей взаимодействия с ней (как правило, из-за голосового управления, требуется оформлять текст по-другому).

## Дополнительно

Здесь находятся параметры, не влияющие на работу чат-бота, но предоставляющие дополнительный функционал.

- **Дополнительные параметры** – это таблица, куда можно добавить любые пары ключ и значение. Их смысл динамически задавать константы, которые могут понадобиться в сценариях. Это позволяет легко их менять прямо во время работы, не изменяя код. Можно придумать следующие области применения:
  - Указания максимального кол-ва выводимых товаров
  - Указания параметров подключения к другим базам
  - Указания вида цен или скидок, при выполнении определенных параметров
  - И т.д.

### **ВНИМАНИЕ:**

- Для переопределения настроек работы в сценарии из поставки «Прайс-лист», в разделе «Дополнительные параметры» можно установить следующие параметры:

*Ключ - ТоварыКолвоНаВывод, Параметр1 – {Число строчек для вывода}*


*Ключ - ТоварыТипЦен, Параметр1 – {Ваш вид цен, допустим Розничная}*

## 3.2. СЦЕНАРИИ ПОВЕДЕНИЯ ЧАТ-БОТА

Сценарии – это описание правил поведения какой-то одной функциональности чат-бота, допустим – создание заказа, отображение баланса, опрос и т.д.

Сценарий универсальный, не привязывается к какой-либо системе чат-ботов и может использоваться в сколько угодно количестве чат-ботов одновременно. Однако для некоторых систем, из-за принципа их работы, рекомендуется создавать отдельные сценарии (допустим – Яндекс.Алиса).

## Список сценариев

Данную форму можно открыть как из подсистемы Чат-боты -> Сценарии чат-бота, так и из мессенджера, нажав на кнопку .

В списке отображается наименование сценария и комментариев. Справочник поддерживает иерархическое представление, с отображением групп. Рекомендуем этим пользоваться, для более удобного расположения элементов. Как правило разделяют по следующим группам: Меню, базовые, функциональность.

В командной панели доступна кнопка «Импортировать сценарии», с помощью нее можно импортировать множество выбранных сценарий автоматически.

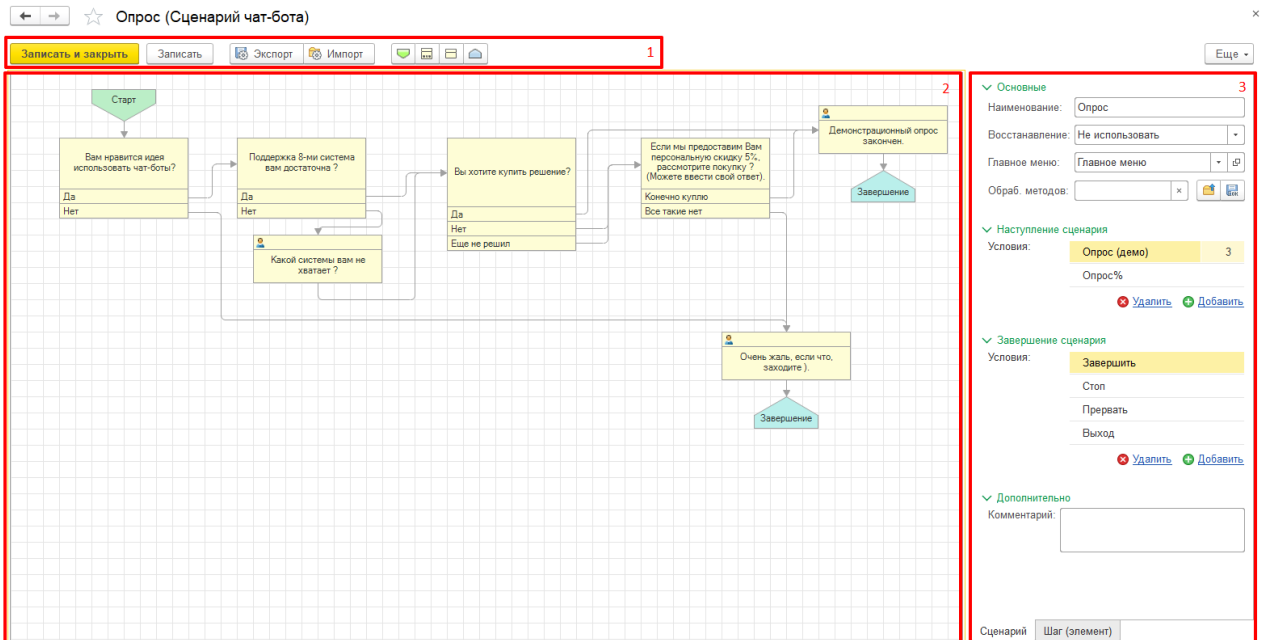
### 3.3. ГРАФИЧЕСКИЙ РЕДАКТОР СЦЕНАРИЕВ (ВЕРСИЯ ENTERPRISE)

Для версии Enterprise, сценарий описывается в визуальном редакторе, с помощью блок схем и установки свойств для каждого шага. Это очень мощное средство, которое позволяет не задумываться над ведением клиента по шагам, отката шагов, проверок и т.д., при этом редактор реализован таким образом, что позволяет описать практически любое поведение чат-бота. Эта целевая схема создания сценариев, она позволяет сократить время разработки сценария с 40-60 чел./часов до 3-5 чел./часов.

Редактор сценария открывается при открытии элемента из списка сценариев или создания нового сценария. В нем происходит основная работа описания поведения бота. Так же в нем происходит подключение обработок расширений и указание методов.

Для версии API данный редактор не поставляется, и создание взаимодействия реализовывает разработчик, используя для отправки данных внутреннее API. Описание этого смотрите в разделе [Руководство разработчика](#).

Рассмотрим подробнее каждую область формы:

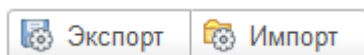


1. Область **Командная панель** – здесь располагаются основные элементы управления сценарием:





**Записать** – после нажатия все данные записываются в базу и, если чат-бот запущен, изменения сразу отобразятся на поведении чат-бота.



**Импорт/Экспорт** – через данные функции можно сохранить все изменения во внешний файл, для передачи его или архивирования. Или загрузить данные, при этом все настройки, блок схемы и т.д. заменяться, включая название сценария.



**Элементы блок схемы** – В данной области расположены все элементы, с помощью которых строятся сценарии. Каждая из кнопок добавляет элемент на схему: Начало, Отправка данных с клавиатурой, Отправка данных, Завершение. Подробнее о них описано ниже.

2. Область **Редактор блок-схемы** – Эта область представляет собой визуальный редактор, посредством стандартных блок-схем. Все возможные элементы доступны из командной панели.
3. Область **Свойства Сценария / Шага** - в данной панели находятся основные свойства, которыми настраивается сценарий и элемент шага. Переключение между свойствами сценария и свойствами шага происходит как автоматически (если выделен элемент, то отображаются свойства шага, если нет выделенного элемента – то свойства сценария), так и вручную на переключении вкладок, внизу панели.

### Создание/редактирование сценария

Основная суть создания сценария – добавление шагов и связей между ними, где шаг – это элемент на форме. Шаг наступает при отправке сообщения пользователем и в нем указывается действия, которые будут выполнены платформой. Элементы Начало и Завершение – не являются шагами, и не могут производить действия, они используются как управляющие / вспомогательные элементы.

Основные правила при создании схемы:

- Элементы обязательно должен быть соединен линиями связи.
- От элемента (или варианта кнопки) должна отходить только одна линия связи.
- В элемент могут приходить неограниченное число связей.

Необходимо учесть, что:

- При копировании элемента, его свойства не копируются.
- При откате действия, свойства не возвращаются.

Создания сценария начинается с указания основных его свойств, они отображаются в панели свойств, если в редакторе блок схем не выбран ни один элемент. И так:

**Свойства сценария** – содержит основные настройки сценария:

**Наименование** – Наименование (представление) самого сценария.

**Условия наступления сценария** – Описывает слова/фразы активизации сценария. Между строчками действует логическое «ИЛИ». С этими условиями сравнивается как сообщение от клиента, так и наименование нажатой кнопки. Здесь допустимы некоторые регулярные выражения:

- % - произвольное кол-во символов. Т.е. слово обрамленное этим знаком будет означать что при любом его вхождении правило будет считаться подобранным. Часто используется указание символа в конце слова
- \_ - один произвольный символ. Практически не используется
- [...] – символы из списка. Практически не используется

Для каждого условия доступно ранжирование, т.е. указание веса данного правила/слова. В итоге, если будет подобрано несколько сценариев, подбирается тот, у которого в сумме вес слов больше.

#### ✓ Наступление сценария

Условия:	Опрос (демо)	10
	Опрос%	1

**Условия завершения сценария** – В списке этой группы указываются ключевые слова, по которым будет происходить безусловный выход из сценария в главное меню. Как правило он используется если клиент на каком-то из шагов решил отказаться от дальнейшего взаимодействия (доступ отправки номера телефона и т.д.). С большей долей вероятности клиент использует слова: выход, завершить, стоп, меню... Проверка работает на равенство слов, без учета регистра.

**Восстанавливать шаг** – Признак, устанавливает возможным продолжение ранее работающего сценария при переключении пользователя на другой сценарий.

*Пример: Пользователь проходил опрос, остановился на каком-то из шагов, далее решил выйти в главное меню, и через некоторое время обратно зашел в сценарий опроса. При значении поля «При прерывании сценария» - опрос продолжится с того места, на котором шаг завершился.*

**Оповестить сотрудника** – Признак, устанавливает автоматическое оповещение сотрудника, если клиент начнет взаимодействовать по данному сценарию.

**Обр. методов** – В данном поле можно загрузить внешнюю обработку (см. раздел [Методы расширяющие возможности сценариев](#)). Экспортные методы модуля этой обработки будут доступны для выполнения в сценарии (формирование сообщения, клавиатуры, подключения событий Перед шагом, проверка ответа и после шага).

После установки свойств основных настроек сценария, можно приступить к выстраиванию шагов и связей между ними, это происходит с помощью элементов блок схем.

#### Элементы блок схем:

**Начало** – элемент, обозначающий начало сценария, с небольшим исключением.

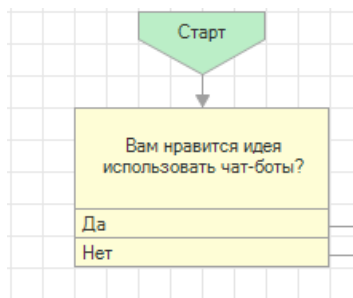


Он может использоваться в 2-х вариантах:

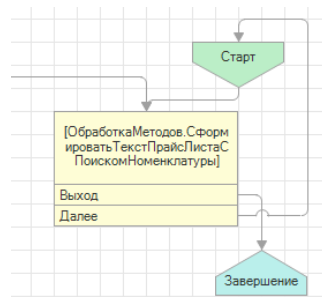
1. Для указания точки начала сценария, при этом от него должна исходить только одна линия связи, к нему не допускается ни одного соединения.

2. Для связи элемента с самим собой, т.е. выход кнопки зациклить на вход в данный шаг, она реализуется через данный элемент.

Пример:

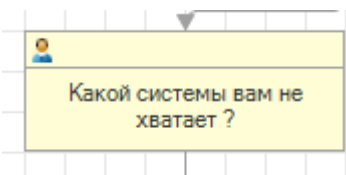


Вариант использования 1.

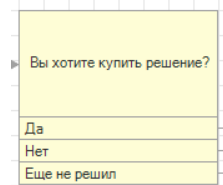


Вариант использования 2.

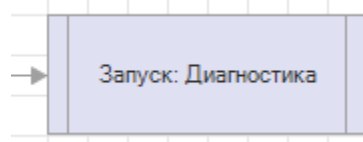
**Отправка данных** – Этот элемент описывает действия, которые платформа выполнит при наступлении шага. Здесь указываются отправка таких действий, как: сообщение, картинка, аудио, видео, файл, контакт и местоположение, запрос номера телефона, запрос местоположения. Данный элемент линейный, его следует использовать если от пользователя ожидается какая-то информация, введенная вручную, т.е. запрос ФИО, возраста, адреса доставки, комментариев и т.д.



**Отправка данных с клавиатурой** – Этот элемент позволяет выполнять такие же действия как «Отправка данных», только дополнительно можно прикреплять клавиатуру/меню, которое отобразится пользователю. От этого элемента будет выходить столько связей, сколько кнопок, т.е. данный элемент реализует разветвление сценария в зависимости от ответа пользователя. Добавление кнопок происходит из панели свойств.



**Вызов сценария** – Этот элемент позволяет запустить произвольный сценарий. Если ссылка на сценарий не указана – то запускается механизм подбора сценария (по условиям наступления), по последнему сообщению, пользователя. Этот шаг должен быть завершающим, после него располагается элемент «Завершение».



**Завершение** – Этот элемент используется для указания завершения сценария, к нему может идти связь с любого элемента, данных элементов может быть сколько угодно. После завершения шага, связь которого идет к нему – сценарий будет завершен и ответ от пользователя не будет ожидаться.



После добавления элемента шага («Отправка данных» или «отправка данных с клавиатурой»), на панели свойств отобразятся поля, специфичные для этого типа элемента.

**Свойства шага (элемента)** – Эта панель содержит описания поведения шага, такие как отправка сообщений, меню, оповещения сотрудника и подписка на события. Остановимся подробнее на каждом из полей:

**Шаг** – Содержит имя шага, по умолчанию имя генерируется, однако, если к нему на каком-то из шагов придется обращаться (допустим для получения ответа на этот шаг) – то лучше указать осмысленное имя или номер шага. Все шаги с ответом далее будут доступны в виде соответствия ключ-значение.

При переименовании шага обязательно следить за тем, что имя не повторяется, иначе сценарий будет вести себя непредсказуемо!

**Отправляемые данные** – Здесь описываются сообщения, данные для отправки пользователю. Возможные все поддерживаемые типы сообщений, а именно:

1. Сообщение – отправляет текстовое сообщение
2. Картинка, Аудио, Видео, Файл – отправляет данные клиенту в необходимом формате, это картинка, аудио или видео – то сразу с отображением и возможностью воспроизвести. Если файл – то скачать.
3. Контакт – отправка номера телефон клиенту с отображением как контакт.
4. Местоположение – отправка местоположения клиенту с отображением на карте. Удобно для уточнения расположения вашего офиса и т.д.
5. Запрос номера телефона – Для авторизации клиента можно запросить у него номер телефона, привязанный к аккаунту (для Телеграмм, вайбер и WhatsApp, для остальных происходит запрос обычным текстовым сообщением).
6. Запрос местоположения – возможность запросить у клиента местоположения (для Телеграмм, вайбер и WhatsApp, для остальных происходит запрос обычным текстовым сообщением).

Если значение в поле «Формирование» оставить по умолчанию, т.е. «**Фиксированное**» то указываются статические данные сообщения. Однако, есть возможность передать управление формированию сообщения в внешнюю функцию. Для этого надо установить значение равное «**По методу**» и в отобразившемся поле указать метод, который будет формировать данные для отправки.

Метод можно указать несколькими способами, подробнее ниже, в пункте «Правила указание динамических методов».

К методу предъявляются требования по сигнатуре параметров (принимаемым параметрам и возвращаемому значению), подробнее можно посмотреть в [Руководство разработчика](#).

**Прикрепляемая клавиатура** – В данной таблице добавляются кнопки, которые выводятся клиенту после отправки сообщения. Первый столбец - № это номер строки вывода клавиатуры (**Не путать с номером кнопки**), он должен начинаться с 1 и может повторяться на нескольких кнопках (на одной строке могут располагаться до 4х кнопок). Увеличивать обязательно последовательно (т.е. не допускается 1, 3, пропустив 2). На каждую кнопку появится вариант в элементе на схеме, его необходимо будет связать с следующим по логике шагом.

Кнопки бывают 2х видов – **меню** (показываются вместо клавиатуры) и **инлайн** (показываются в тексте). Инлайн кнопки поддерживаются не всеми системами, если не поддерживается – то вывод эмулируется текстом.

При показе кнопок, ответ клиента проверяется на корректность соответствие кнопкам, т.е. ответ должен быть дан либо по нажатию кнопки, либо введено ее

№	Текст
1	По логину/паролю
1	По номеру телефона

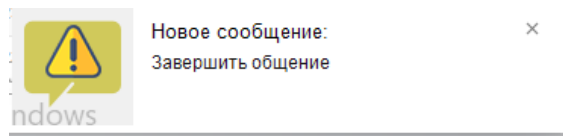
название. Если необходимо отключить данный механизм – то следует установить галочку «**Не проверять корректность ответа по кнопкам**», при этом на элементе добавиться еще один вариант «...» по нему пойдет сценарий при альтернативном ответе.

**Изменить данные отправки динамически** - Для более гибкого создания данных отправки и кнопок, есть механизм «Изменить данные отправки динамически». Основная цель его – дать возможность динамически кодом изменять состав данных отправки сообщений и кнопок. В данном методе есть возможности добавить, изменить и удалить как сообщения любых типов, так и кнопки. Для кнопок есть ограничение, все варианты связей должны быть прописаны заранее в графическом конструкторе, при добавлении кнопки наименование указывается произвольное, а ключ – существующая кнопка со связью. Подробнее можно посмотреть в пункте «Правила указание динамических методов» и в [Руководство разработчика](#).

Данный механизм, изменение сообщений может применяться, для формирования товаров, их всегда разное кол-во и в конструкторе невозможно угадать их число. Изменение кнопок применяется в различных целях, допустим какому-то пользователю необходимо предоставить дополнительные возможности, или убрать их, или допустим для формирования номера страницы в кнопки (страница 3 из 10) и т.д.

**Аккумулировать ответы шага в массиве** – Это очень важный параметр, который позволяет в контексте ответов, все ответы пользователя на текущем шаге, в соответствии с их порядком ответа. Если галочка не установлена – то на шаге будет возвращена структура с единственным ответом, если их было несколько – возвращается последний ответ. Использовать этот механизм можно к примеру, в сценарии покупки, когда клиент вводит артикул интересующего товара, и далее может еще раз добавить другой товар, в таком случае мы увидим множество ответов (в контексте будет фиксированный тип массив, в случае если галочка не стоит – то только Структура).

**Создать оповещение сотруднику** – Этот признак устанавливается при необходимости оповещения сотрудника об ответе на шаг. Сообщение пользователя будет отображено на рабочем столе 1С (при открытом мессенджере), при нажатии на которое будет переход в диалог взаимодействия с клиентом.



**События** – Это триггеры, при наступлении которых, в зависимости от стадии шага, выполняется указанная процедура/функция. Это очень мощный функционал. Он позволяет: проводить дополнительные проверки ответа (допустим сравнивая с данными из БД); Вызывать метод создания заказа, сделки; Опишем подробнее виды событий:

1. **Перед шагом** – Событие срабатывает перед запуском текущего шага, т.е. перед выполнением каких-либо действий этого шага.

Может применяться механизм пропуска шага, для этого в возврате функции необходимо вернуть структуру с свойством «ПропуститьШаг».

**Примечание:** Данный механизм применяется редко, как правило только для пропуска авторизации.

2. **Проверка ответа** – Событие срабатывает при получении ответа от пользователя на текущий шаг. Может применяться для проверки корректности ответа (пример: корректность номер договора, номера телефона, почты и т.д.). Возможно как запрос повтора ввода ответа, так и принудительное прекращение текущего сценария (Пример: договор не прошел проверку, клиент в черном списке и т.д.)  
**Примечание:** Не выполняется на последнем шаге сценария, т.к. последний шаг завершающий, без ожидания ответа.

3. **После шага** – Событие срабатывает при завершении шага. Это событие может применяться в основном для 2х задач:

- Выполнение завершающих действий шага (пример: создание итогового документа, справочника и т.д.)

- Для принудительной установка следующего шага, по какому-то условию.

**Примечание:** Не выполняется на последнем шаге сценария (будет исправлено).

Метод можно указать несколькими способами, подробнее ниже, в пункте «Правила указание динамических методов».

К методу предъявляются требования по сигнатуре параметров (принимаемым параметрам и возвращаемому значению), подробнее можно посмотреть в

[Руководство разработчика](#)

## Правила указания динамических методов:

Метод может быть указан 2-я вариантами:

1. [ИмяОбработки].ИмяЭкспортногоМетода

Обработка, может быть загружена в самом сценарии, в поле Дополнительно – обр. методов или через механизм «Дополнительные обработки и отчеты» Целевой вариант использования является загрузка через механизм «Дополнительные обработки и отчеты» - так как он позволяет один раз загрузить обработку и во всех сценариях ее использовать. Механизм загрузки обработки в сценарии не рекомендуется к использованию и применяется в основном только для самописных конфигураций или, иногда, для тестирования.

2. ИмяМодуля.ИмяЭкспортногоМетода

Любой экспортный метод конфигурации (модуля, справочника, документа и т.д.) Вариант для тестирования на демо базе, доступна отладка.

**ВНИМАНИЕ:** Для повышения производительности обработка кэшируется, чтобы, после обновления, она перекэшировалась, необходимо предварительно изменить версию в структуре СведенияОВнешнейОбработке.

# РУКОВОДСТВО РАЗРАБОТЧИКА, API

## 4.1. ПОДКЛЮЧАЕМЫЕ МЕТОДЫ (ВНЕШНЯЯ ОБРАБОТКА)

Графический редактор сценариев – это очень мощный механизм, с помощью которого можно построить даже сложный сценарий за несколько десятков минут. Однако, как правило, при создании сложного сценария, необходим дополнительный механизм, расширяющий возможности управления шагами - динамическое формирование текста, сложных проверок сообщений, итоговое создание документов и т.д. Для этого реализована возможность подключения обработки к сценарию, и последующего выполнения любых экспортных методов.

Обработка должна соответствовать стандартным требованиям внешних подключаемых обработок в 1С. Т.е. в модуле обработке должен быть определен метод с возвращаемой структурой – информацией об обработке:

Функция СведенияОВнешнейОбработке() Экспорт

Результат = Новый Структура();

Результат.Вставить("Вид", "ДополнительнаяОбработка");

Результат.Вставить("Версия", "5.00");

Результат.Вставить("Наименование", "Vots1С: Базовые методы");

Результат.Вставить("БезопасныйРежим", Ложь);

Результат.Вставить("Назначение", Новый ТаблицаЗначений);

Результат.Вставить("Команды", Новый ТаблицаЗначений);

Результат.Вставить("Информация", "ПлатформаИнтеграцииЧатБотов: Обработка, добавляющая методы базовой функциональности для расширения возможности сценарев.");

Возврат Результат;

КонецФункции

Все методы, которые будут доступны для вызова – необходимо объявлять с ключевым словом «Экспорт».

Методы, которые подключены к сценарию, вызываются ядром и у них сигнатура параметров должна точно соответствовать описанным ниже, в зависимости от назначения метода. Теперь остановимся на методах и их назначении, которые можно подключить к сценарию:

### 1. Динамически формируемые данные сообщения для отправки.

Функция ДинамическоеСообщение(ЧатБот, IDЧата, СценарийСсылка, ШагиВзаимодействия) Экспорт

Возврат ..... ;

КонецФункции

Входящие параметры:

**ЧатБот** – <Ссылка> на справочник чат ботов.

**IDЧата** – <Число, Строка> ID пользователя во внешней системе.

**СценарийСсылка** – <Ссылка> на выполняемый сценарий.

**ШагиВзаимодействия** – Соответствие, где ключ – имя шага, а значение структура ответа, из регистра. Если установлена галочка аккумулировать ответ – то массив с элементами такой структуры

Возвращаемое значение: в зависимости от типа сообщения:

Сообщение: <Строка>

Картинка, Аудио, Видео, Файл:

<Структура(«МультимедиаДвоичныеДанные, МультимедиаИмяФайла», <ДвоичныеДанные>, <Строка>)>

Местоположение: <Структура(«Широта, Долгота», <Число>, <Число>)>

Контакт: <Структура(«Телефон, Имя», <Строка>, <Строка>)>

ЗапросМестоположения: <Структура(«Сообщение, ТекстКнопки», <Строка>, <Строка>)>

ЗапросНомераТелефона: <Структура(«Сообщение, ТекстКнопки», <Строка>, <Строка>)>

Данный механизм необходим для динамического формирования данных сообщения (текста, картинки, аудио, видео, файла, номера телефона, контакта, запроса номера телефона и запроса контакта).

Подключение: На любом шаге сценария, в группе «Отправляемые данные», в добавленных данных для отправки необходимо выбрать Формирование – «по методу», и в поле ниже указать путь к методу.

Пример:

```
Функция ДинамическоеСообщение(ЧатБот, IDЧата, СценарийСсылка, ШагиВзаимодействия) Экспорт
    Возврат "Этот шаг тестирует отправку динамических данных";
    Возврат Новый Структура("МультимедиаДвоичныеДанные, МультимедиаИмяФайла",
        ЭтотОбъект.ПолучитьМакет("Демо_Картинка"), "Демо_Картинка.jpg");
    Возврат Новый Структура("Широта, Долгота", 55.751596, 48.750269);
    Возврат Новый Структура("Телефон, Имя", "+7967727515", "Интеграция ботов с 1С");
    Возврат Новый Структура("Сообщение, ТекстКнопки", "Укажите местоположение", "Отправить");
    Возврат Новый Структура("Сообщение, ТекстКнопки", "Укажите номер телефона", "Отправить");
    Возврат Новый Структура("Сообщение, ТекстКнопки", "Укажите ваше местоположение", "Отправить");
    Возврат Новый Структура("Сообщение, ТекстКнопки", "Укажите ваш номер", "Отправить");
КонецФункции
```

## 2. Динамическое изменение данных отправки (сообщений и клавиатуры):

```
Функция ДинамическоеИзменениеСообщенийКлавиатуры (ЧатБот, IDЧата, СценарийСсылка,
ШагиВзаимодействия, ДанныеСообщений, ДанныеКлавиатуры)
    Возврат Структура("ДанныеСообщений, ДанныеКлавиатуры", <МассивСтруктур>, <МассивСтруктур>);
КонецФункции
```

Входящие параметры:

**ЧатБот** – <Ссылка> на справочник чат ботов.

**IDЧата** – <Число, Строка> ID пользователя во внешней системе.

**СценарийСсылка** – <Ссылка> на выполняемый сценарий.

**ШагиВзаимодействия** – <Ссылка> на выполняемый сценарий.

**ДанныеСообщений** – <Элемент массива>: Структура("Тип<Перечисление.ТипыСообщенийЧатБота>,
ПризнакФормироватьПоМетоду<Булево>, ...СвойстваПоТипу..."),
где ПризнакФормироватьПоМетоду - Ложь, если сообщение будет статическое, если далее
вычисляться методом - Истина, и добавляется свойство ПутьКМетоду<Строка>
СвойстваПоТипу - Уникальные свойства для текущего типа данных, можно посмотреть выше в
ДинамическоеФормированиеСообщений(Для сообщение доп.свойство "Сообщение")

**ДанныеКлавиатуры** – <Элемент массива>: Структура("Тип<Перечисление.ТипыСообщенийЧатБота>,
НомерСтроки<Число>, КлючКлавиши<Строка>, Представление<Строка>"),
где НомерСтроки - Номер строки вывода кнопок, должен быть неразрывным, может
повторяться у нескольких кнопок, не более 5-и.
КлючКлавиши - Должен соответствовать существующему ключу из редактора, для
корректности связей.
Представление – Наименование кнопки.

Возвращаемое значение:

```
Структура("ДанныеСообщений, ДанныеКлавиатуры", <МассивСтруктур>, <МассивСтруктур>);
```

Данный механизм позволяет динамически изменить сформированные сообщения и клавиатуру. Т.е. возможно добавить, изменить и удалить какое-либо отдельное сообщение или кнопку клавиатуры. но с указанием ключа для связи со следующим шагом. При этом в сценарии обязательно должен быть этот шаг с клавиатурой, и связями на каждой кнопке.

```
Функция Диагностика_ДинамическоеИзменениеКнопок(ЧатБот, IDЧата, СценарийСсылка,
ШагиВзаимодействия, Клавиатура) Экспорт
    ДанныеСообщений.Добавить(Новый Структура("Тип, ПризнакФормироватьПоМетоду, Сообщение",
Перечисления.ТипыСообщенийЧатБота.Сообщение, Ложь, "Товар №1: ....."));
```



ДанныеСообщений.Добавить(Новый Структура("Тип, ПризнакФормироватьПоМетоду, Широта, Долгота", Перечисления.ТипыСообщенийЧатБота.Местоположение, Ложь, 55.751596, 48.750269));  
ДанныеСообщений[ИндексСообщенияРанееВычисленный].Сообщение = "Страница 5 из 10";  
ДанныеСообщений.Удалить(ИндексСообщенияРанееВычисленный);  
ДанныеКлавиатуры.Добавить(Новый Структура("Тип, НомерСтроки, КлючКлавиши, Представление", Перечисления.ТипыСообщенийЧатБота.КнопкаКлавиатуры, 1, "Дальше", "Последняя страница"));  
ДанныеКлавиатуры[ИндексКнопкиДальшеРанееВычисленный].Представление = "Страница 5 из 10";  
ДанныеКлавиатуры.Удалить(ИндексКнопкиДальшеРанееВычисленный);  
Возврат Новый Структура(«ДанныеСообщений, ДанныеКлавиатуры», ДанныеСообщений, ДанныеКлавиатуры);

КонецФункции

### 3. Подключаемый обработчик события - ПередШагом

Функция Метод\_ПриОтвете(ЧатБот, IDЧата, СценарийСсылка, ШагиВзаимодействия,) Экспорт

Возврат ..... ;

КонецФункции

Входящие параметры:

**ЧатБот** – <Ссылка> на справочник чат ботов.

**IDЧата** – <Число, Строка> ID пользователя во внешней системе.

**СценарийСсылка** – <Ссылка> на выполняемый сценарий.

**ШагиВзаимодействия** – Стандартный параметр, описан выше.

Контракт на возвращаемое значение:

<Структура("ПропуститьШаг ")>

Где,

ПропуститьШаг – <Булево> Если текущий шаг необходимо пропустить.

Может применяться для пропуска текущего шага при каких либо условий.

*Примечание: Данный механизм применяется редко, как правило только для пропуска запроса авторизации.*

### 4. Подключаемый обработчик события - ПроверкаОтвета

Функция Метод\_ПриОтвете(ЧатБот, IDЧата, СценарийСсылка, ШагиВзаимодействия, Запись\_ОтветНаШаг)

Экспорт

Возврат ..... ;

КонецФункции

Входящие параметры:

**ЧатБот** – <Ссылка> на справочник чат ботов.

**IDЧата** – <Число, Строка> ID пользователя во внешней системе.

**СценарийСсылка** – <Ссылка> на выполняемый сценарий.

**ШагиВзаимодействия** – Стандартный параметр, описан выше.

**ОтветНаШаг** – <Структура> все поля записи регистра ИсторияСообщенийЧатБота.

Контракт на возвращаемое значение:

<Структура("Отказ, ЗавершитьСценарийПриОтказе, СообщениеПриОтказе")>

Где,

Отказ – <Булево> Если ответ не прошел проверку и надо запросить повтор ввода.

ЗавершитьСценарийПриОтказе – <Булево> Если ответ не прошел проверку и необходимо завершить работу сценария.

СообщениеПриОтказе – <Строка> Сообщение, которое будет отправлено клиенту.

Может применяться для проверки корректности ответа (пример: корректность номер договора, номера телефона, почты и т.д.). Возможно как запрос повтор ввода ответа, так и принудительное прекращение текущего сценария (Пример: договор не прошел проверку, клиент в черном списке и т.д.)

*Примечание: Не выполняется на последнем шаге сценария, т.к. последний шаг завершающий, без ожидания ответа.*

Пример: Смотрите метод «Диагностика\_ПриОтвете» в обработке БазовыеМетодыЧатБотов.

## 5. Подключаемый обработчик события - ПослеЗавершенияШага

Функция Метод\_ПослеШага(ЧатБот, IDЧата, СценарийСсылка, ШагиВзаимодействия) Экспорт  
Возврат ..... ;  
КонецФункции

Входящие параметры:

- ЧатБот** – <Ссылка> на справочник чат ботов.
- IDЧата** – <Число, Строка> ID пользователя во внешней системе.
- СценарийСсылка** – <Ссылка> на выполняемый сценарий.
- ШагиВзаимодействия** – Стандартный параметр, описан выше.

Контракт на возвращаемое значение:

<Структура("УстановитьСледующийШаг")>  
Где, УстановитьСледующийШаг – <Строка> Имя шага, как он указан в сценарии.  
Или Неопределенно, в случае если не требуется изменить шаг.

Это событие может применяться в основном для 2х задач:

- Выполнение завершающих действий шага (пример: создание итогового документа, справочника и т.д.)
- Для принудительной установка следующего шага, по какому то условию.

## 4.2. API

Все методы взаимодействия с внешними системами чат ботов – унифицированы и приведены к одному общему виду и выделены в отдельный API, скрывающей всю внутреннюю реализацию транспорта, для более легкого взаимодействия с чат-ботами. Т.е. для отправки сообщений существует всего один метод, и одним из параметров передается идентификатор системы чат-ботов. Это упрощает использование и избавляет от описания множества вариаций использования одного метода в зависимости от системы.

«Ручная» отправка сообщений может пригодиться:

- При создании оповещений клиента при определенных действиях, в обработке подписки на события
  - При создании оповещений клиента о задолженностях, в регламентных заданиях
  - При создании розыгрышей, когда розыгрыш стартует в определенное время, и необходимо инициировать первое сообщение с запуском сценария
- И т.д.

Взаимодействие с чат ботом:

### 1. Функция ЧатБотAPI.ОтправитьСообщение(ЧатБот, IDЧата, Сообщение, Клавиатура, Настройки)

Данная функция необходима для отправки сообщений, последним параметром можно передать клавиатуру для вывода.

Где параметры:

- ЧатБот** – <Ссылка> на справочник чат ботов.
- IDЧата** – <Число, Строка> ID пользователя во внешней системе.
- Сообщение** – <Строка>, отправляемое сообщение.
- Клавиатура** – <Массив<Структур>> Клавиатура, отображаемая пользователю, описан ниже.
- Настройки** – <Структура> Этот параметр содержит дополнительные настройки отправки сообщений, он не является обязательным. Возможные свойства:
  - УстановитьСценарий**– <Ссылка.СценарииЧатБотов> При передаче данного значения, при записи в историю сообщений запишется данный сценарий, цель его указания – направить взаимодействие по определенному сценарию.
  - УстановитьОжиданиеОтвета**– <Булево> Признак, при указании которого будет ожидаться ответ пользователя, после ответа управление передается ядру с целью подбора дальнейшего шага внутри активного сценария.
- Статус** – <Булево> Признак, показывающий результат отправки (Удачно или ошибка отправки)
- IDЗаписи** – <УникальныйИдентификатор> ID записи в регистре сведений ИсторияСообщений.
- ОписаниеОшибки** – <Строка> Описание ошибки

*Пример:*

*Результат = ЧатБотAPI.ОтправитьСообщение(ЧатБотСсылка, "1861315", "Ваш текст сообщения");*

## **2. Функция ЧатБотAPI.ОтправитьДвоичныеДанные(ЧатБот, IDЧата, ТипДанных, ТипДанных, ДвоичныеДанные, ИмяФайла, Клавиатура, Настройка)**

С помощью данной функции можно отправить мультимедиа-данные: картинки, аудио, видео или произвольные файлы.

Где параметры:

**ЧатБот, IDЧата, Клавиатура, Настройка** – см. в пункте 1.

**ТипДанных** – <Перечисления.ТипыСообщенийЧатБота> Тип отправляемых данных:

Перечисления.ТипыСообщенийЧатБота.Картинка

Перечисления.ТипыСообщенийЧатБота.Аудио

Перечисления.ТипыСообщенийЧатБота.Видео

Перечисления.ТипыСообщенийЧатБота.Файл

**ДвоичныеДанные** – <ДвоичныеДанные> Данные мультимедиа файла

**ИмяФайла** – <Строка>, тестовое имя файла с расширением. Пользователь увидит его при скачивание, или при отображении файла – гиперссылкой.

Возвращаемое значение: см. в пункте 1.

*Пример:*

*Результат = ЧатБотAPI.ОтправитьДвоичныеДанные(ЧатБотСсылка, "1861315",  
Перечисления.ТипыСообщенийЧатБота.Картинка, ДвоичныеДанныеФайла);*

## **3. Функция ЧатБотAPI.ОтправитьКонтакт(ЧатБот, IDЧата, НомерТелефона, Представление, Клавиатура, Настройка)**

Функция предназначена для отправки сообщения в виде контакта, т.е. с номером телефона и представлением.

Где параметры:

**ЧатБот, IDЧата, Клавиатура, Настройка** – см. в пункте 1.

**НомерТелефона** – <Строка> Номер телефона в международном формате.

**Представление** – <Строка> Представление контакта.

Возвращаемое значение: см. в пункте 1.

*Пример:*

*ЧатБотAPI.ОтправитьКонтакт(ЧатБотСсылка, "1861315", "+7xxxxxxxx", "Боты в 1с");*

## **4. Функция ЧатБотAPI.ОтправитьМестоположение(ЧатБот, IDЧата, Широта, Долгота, Клавиатура, Настройка)**

Данная функция позволяет отправить координаты с отображением их на карте.

Где параметры:

**ЧатБот, IDЧата, Клавиатура, Настройка** – см. в пункте 1.

**Широта** – <Число> Широта, в виде xx.xxxxxx

**Долгота** – <Число> Долгота, в виде xx.xxxxxx

Возвращаемое значение: см. в пункте 1.

*Пример:*

*ЧатБотAPI.ОтправитьМестоположение(ЧатБотСсылка, "1861315", 55.7537926, 48.7433563);*

## **5. Функция ЧатБотAPI.ЗапроситьНомерТелефона(ЧатБот, IDЧата, Сообщение, ТекстКнопки, Настройка)**

Данная функция используется для запроса номера телефона клиента, прикрепленного к аккаунту, ответ сохраняется в истории. Может быть с разным типом данных: ПодтвержденныйКонтакт – если была нажата кнопка и отправлен контакт на аккаунте или Контакт - если номер телефона был введен вручную как сообщение. Таким образом производится идентификация клиента и связка его с контрагентами из справочника Контрагенты.

Где параметры:

**ЧатБот, IDЧата, Настройка** – см. в пункте 1.

**Сообщение** – <Строка> Сообщение с запросом номера телефона, отправляемое пользователю.

**ТекстКнопки** – <Строка> Текст кнопки отправки номера телефона (если бот поддерживает).

Возвращаемое значение: см. в пункте 1.

*Пример:*

*ЧатБотAPI.ЗапроситьНомерТелефона(ЧатБотДанные, "1861315", "Укажите ваш номер телефона для авторизации (в полном формате +...)", "Отправить номер телефона");*

## 6. Функция ЧатБотAPI.ЗапроситьМестоположение(ЧатБот, IDЧата, Сообщение, Настройка)

Данная функция используется для запроса местоположения клиента. Специфическая функция, обычно не используется, ответ сохраняется в истории.

Где параметры:

**ЧатБот, IDЧата, Настройка** – см. в пункте 1.

**Сообщение** – <Строка> Сообщение с запросом местоположения, отправляемое пользователю.

**ТекстКнопки** – <Строка> Текст кнопки отправки местоположения (если бот поддерживает).

Возвращаемое значение: см. в пункте 1.

*Пример:*

*ЧатБотAPI.ЗапроситьМестоположение(ЧатБотДанные, "1861315", "Укажите ваше местоположение", "Отправить местоположение");*

Работа с сценариями:

## 7. Процедура ЧатБотAPI.ЗапуститьСценарий(ЧатБотСсылка, IDЧата, СценарийСсылка, ШагСценария = Неопределено)

Необходима для запуска сценария, может применяться допустим в регламентном задании, при каких то условиях – запускать сценарий для взаимодействия с клиентом.

Где параметры:

**ШагСценария** – <Строка>, необязательный параметр. Указывает шаг сценария с которого запустить сценария. По умолчанию сценарий запускается сначала.

*Пример:*

*Значение = ЧатБотAPI.ЗапуститьСценарий (ЧатБотДанные, "1861315", Сценарий, «4»)*

## 8. Процедура ЧатБотAPI.ЗаписатьДанныеВКонтекст(ЧатБотСсылка, IDЧата, СценарийСсылка, КлючШаг, Данные = Неопределено)

Данный метод позволяет записать какие-то временные данные в контекст сценария, допустим авторизацию, или номенклатуру которую клиенту выводить, определяя ее в другом шаге.

Где параметры:

**КлючШаг** – <Строка> Ключ данных

**Данные** – <Строка> Данные в строке, если необходимо записать какую то таблицу, массив и т.д. можно предварительно воспользоваться функцией ЗначениеВСтрокуВнутри()

*Пример:*

*Контрагент = ЧатБотAPI.ЗаписатьДанныеВКонтекст (ЧатБотДанные, "1861315", Сценарий, «НоменклатураНаВывод», ЗначениеВСтрокуВнутри(ТаблицаНоменклатуры))*

## 9. Функция ЧатБотAPI.ПолучитьКонтекстВзаимодействия (ЧатБотСсылка, IDЧата, СценарийСсылка, НаПериод = Неопределенно)

Данная функция является обратной к ЗаписатьДанныеВКонтекст, и позволяет получить записанные ранее данные.

## 10. Процедура ЧатБотAPI.ПолучитьПользовательскийПараметрЧатБота (ЧатБотСсылка, КлючПараметра, ЗначениеПоУмолчанию = Неопределено)

Служит для получения произвольных параметров, которые указываются в элементе чат-бота. Допустим кол-во сообщений для вывода и т.д., которые некорректно было бы прописывать в коде методов.

Где параметры:

**ЧатБотСсылка** – см. в пункте 1.

**КлючПараметра** – <Строка> Ключ параметра из таблицы с параметрами

**ЗначениеПоУмолчанию** – Если значение не найдено, какое значение выводить, позволяет избежать лишнего дальнейшего условия

*Пример:*

*ЧатБотAPI.ПолучитьПользовательскийПараметрЧатБота (ЧатБот, "ВидЦен", Справочник.ВидыЦен.Розница)*

Работа с контактами:

## 11. Функция ПолучитьКонтактЧата(ЧатБотСсылка, IDЧата)

По данным чата получает структуру с данными контакта, где его наименование, номер телефона, контрагент и т.д.

Где параметры:

**ЧатБот, IDЧата** – см. в пункте 1.

Возвращаемое значение: <Структура>

**Подписан** – <Булево> Признак подписан ли клиент на бота, не всегда может быть корректным

**Представление** – <Строка> Представление пользователя, может быть Имя и Фамилия или логин

**НомерТелефона** – <Строка> Номер телефона

**СсылкаНаАватар** – <Строка> URL ссылка на картинку аватара

**Контрагент** – <Ссылка.Контрагенты> Ссылка на контрагента, если удалось связать

*Пример:*

*Контакт = ЧатБотAPI.ПолучитьКонтактЧата(ЧатБот, "1861315")*

## **12. УстановитьНомерТелефонаКонтакта(ЧатБотСсылка, IDЧата, НомерТелефона, КонтрагентСсылка = Неопределено)**

Необходим для «ручной» установки данных контакта

Где параметры:

**ЧатБот, IDЧата** – см. в пункте 1.

Возвращаемое значение: <Структура>

**НомерТелефона** – <Строка> Номер телефона

**Контрагент** – <Ссылка.Контрагенты> Ссылка на контрагента, если передать Неопределенно – будет произведен поиск контрагента методом **ПолучитьКонтрагентаПоНомеруТелефона**

*Пример:*

*Контакт = ЧатБотAPI.УстановитьНомерТелефонаКонтакта (ЧатБот, "1861315", «89677727515»)*

Механизмы сотрудника:

## **12. Процедура ОповеститьСотрудника(ЧатБотСсылка, IDЧата, IDЗаписиСообщения, ИдентификаторПользователяИБ = Неопределено, ВидОповещения = "", Примечание = "")**

Создает оповещение сотруднику, при нажатии на которое будет переход в мессенджер.

Где параметры:

**ЧатБот, IDЧата** – см. в пункте 1.

Возвращаемое значение: <Структура>

**IDЗаписиСообщения** – <УникальныйИдентификатор> ID сообщения

**ИдентификаторПользователяИБ** – <УникальныйИдентификатор> ID пользователя, которому отправить оповещение, если будет Неопределенно – то оповещение будет всем.

**ВидОповещения** – <Строка> Произвольная строка, по умолчанию пусто (лучше так оставлять)

**Примечание** – <Строка> При указании эти данные будут отображены сотруднику в оповещении

*Пример:*

*Контакт = ЧатБотAPI.ПолучитьКонтактЧата(ЧатБот, "1861315")*

Методы базовой конфигурации:

## **13. Функция ОпределитьПользователяДляОповещения(ЧатБот, IDЧата, Сценарий = Неопределено)**

Метод, в котором реализован механизм назначения сотрудника для показа оповещения о получении сообщения от клиента. По умолчанию работает следующий механизм: Ищется пользователь последнего общения; ищется основной менеджер клиента из карточки контрагента/партнера.

Если необходим иной механизм – переопределить его можно, скопировав данный метод, и указав к нему путь в чат-боте в пользовательских параметрах, выбрав ключ

«\_ОпределитьПользователяДляОповещения»

## **14. Функция ПолучитьКонтрагентаПоНомеруТелефона(НомерТелефона, ЧатБот = Неопределено)**

Метод, в котором описываются запросы для получения контрагента/партнера в БД по номеру телефона. По умолчанию запросы адаптированы к конфигурациям УТ, УНФ, Розница, CRM. Если конфигурация иная – можно скопировать функцию и адаптировать запрос, при этом указав к нему путь в чат-боте в пользовательских параметрах, выбрав ключ «\_ПолучитьКонтрагентаПоНомеруТелефона»