

Документация





ООО «Грин-АПИ»

ИНН 5047259512/КПП 504701001

ОГРН 1215000131532

Адрес: Россия, 141402, Московская область, г. Химки, ул. Московская, стр. 38А, помещ. 9/001, БЦ "Панорама"

Тел.: +7 (999) 333-12-23

Почта: support@green-api.com

Сайт: <https://green-api.com>

Документация
Программа для ЭВМ
«GREEN-API»

1. Документация

1.1 Оглавление

Table of contents

1. Документация	4
1.1 Оглавление	4
1.2 Быстрый старт	5
1.2.1 Перед началом работы	5
1.2.2 Тарифы	9
1.2.3 Выполнение запросов	14
1.2.4 Коллекция Postman	15
1.3 Документация API	17
1.3.1 Документация API	17
1.3.2 Аккаунт	20
1.3.3 Отправка	40
1.3.4 Получение	60
1.3.5 Журналы	131
1.3.6 Очереди	168
1.3.7 Группы	174
1.3.8 Статусы (бета-версия)	187
1.3.9 Отметка прочтения	204
1.3.10 Сервисные методы	207
1.3.11 Прочее	222
1.3.12 Ограничение частоты запросов	229
1.3.13 Рекомендации по интеграции	233
1.4 Личный кабинет	244
1.4.1 Личный кабинет	244
1.4.2 Регистрация в личном кабинете	245

1.2 Быстрый старт

1.2.1 Перед началом работы

Перед началом работы с Green API требуется выполнить следующие шаги:

1. [Установить мобильное приложение WhatsApp](#)
2. [Регистрация в Личном кабинете](#)
3. [Создание и авторизация инстанса](#)
4. [Получить параметры доступа к инстансу](#)
5. [Настроить получение входящих данных](#)

1. Установить мобильное приложение WhatsApp

Отправка и получение сообщений WhatsApp выполняется с вашего мобильного телефона. На телефоне должно быть установлено официальное мобильное приложение [WhatsApp](#) или [WhatsApp Business](#).

"Выбор приложения"

Выбор приложения WhatsApp или WhatsApp Business не влияет на вероятность блокировки аккаунта.

Рекомендуется заказать услугу **Хостинг трубки**. В этом случае наличие личного мобильного телефона не требуется. Отправка и получение сообщений WhatsApp будет выполняться с мобильного телефона, расположенного в дата-центре Green API.

2. Регистрация в Личном кабинете

Для использования нашего сервиса перейдите в [Личный кабинет](#) и зарегистрируйтесь. Порядок регистрации:

1. Введите электронную почту.
Рекомендуем использовать корпоративную почту, для предотвращения потери доступа к учетной записи, при увольнении сотрудника.
2. Введите пароль и подтвердите его. Длина пароля должна быть не менее 6 символов.
3. Выберите страну.
4. Приняв пользовательское соглашение, нажмите на кнопку [Зарегистрироваться](#).
5. На Ваш адрес электронной почты будет направлено письмо. Введите код, указанный в письме, для подтверждения вашей учетной записи.

3. Создание и авторизация инстанса

СОЗДАНИЕ ИНСТАНСА

Инстанс - это уникальный номер шлюза для отправки и получения сообщения через WhatsApp.

Инстанс создается в [личном кабинете](#) и используется для организации HTTP API WhatsApp.

Один инстанс может одновременно обслуживать только один номер телефона (один аккаунт WhatsApp). В одной учетной записи можно создать несколько инстансов.

Порядок создания инстанса:

1. В [личном кабинете](#) нажмите на кнопку [Создать инстанс](#).
2. Выберите тарифный план
3. Если требуется, выберите способ оплаты и проведите оплату.
4. Перейдите в список инстансов, нажав на кнопку [Инстансы](#) в боковом меню. В списке можете увидеть созданный вами инстанс.

Переход инстанса в рабочее состояние после создания может занимать до 2-х минут. Формирование QR кода возможно через 2 минуты после создания инстанса.

АВТОРИЗАЦИЯ ИНСТАНСА

Для работы с Green API требуется авторизовать инстанс. Авторизация инстанса выполняется в [Личном кабинете](#) путем считывания QR-кода из мобильного приложения [WhatsApp Business](#) или [WhatsApp](#).

Порядок авторизации инстанса:

1. Откройте на своем мобильном телефоне приложение **WhatsApp Business** или **WhatsApp**. На устройстве перейдите в раздел Привязка устройства, для этого:
 - На Android нажмите на 3 точки -> Связанные устройства -> Привязка устройства
 - На iPhone перейдите в Настройки -> Связанные устройства -> Привязка устройства
2. Перейдите в **личный кабинет**. Выберите нужный вам инстанс.
3. Нажмите на кнопку **Получить QR**. Отсканируйте QR код.

GREEN API

Изменить Разлогинить Перезапустить Удалить

Тариф BUSINESS
Инстанс оплачен до сб, 03 февр. 2024 г.

API url	https://api.green-api.com
Media url	https://media.green-api.com
IdInstance	1101
ApiTokenInstance	793450914c78464fbe209bf
Наименование	Instance 1101
Статус	Неавторизован
Телефон	

Инстансы
API
Баланс
Покупки
Профиль
Документация

Нажмите кнопку получить Qr код, чтобы появилась картинка с актуальным Qr

Получить QR

WhatsApp Business

Business tools

- New group
- New broadcast
- Labels
- Linked devices
- Stored messages

GREEN-API ©2023

Отправка и получение сообщений выполняется с вашего мобильного телефона. Поэтому телефон должен быть всегда заряжен и подключен к сети Интернет.

4. Получить параметры доступа к инстансу

Для выполнения запросов HTTP API WhatsApp требуется использовать параметры доступа к инстансу. Параметры доступа публикуются в **личном кабинете**:

- `apiUrl` - ссылка на хост API
- `mediaUrl` - ссылка на хост API для отправки файлов
- `idInstance` - уникальный номер инстанса
- `apiTokenInstance` - ключ доступа инстанса

Ключ доступа инстанса можно сменить при необходимости, например, в случае его компрометации. Сменить токен можно в [личном кабинете](#) самостоятельно или обратившись в [техподдержку](#).

5. Настроить получение входящих данных

Если требуется получать входящие данные (входящие сообщения, статусы ранее отправленных сообщений и др.), то предварительно необходимо выполнить [настройку инстанса](#). Получать входящие уведомления можно [через HTTP API](#) или [через Webhook Endpoint](#).

Готово!

Всё готово для начала отправки и получения сообщений WhatsApp!

Для отладки запросов к Green API рекомендуется использовать [раздел API в личном кабинете](#) или [Коллекцию Postman](#).

1.2.2 Тарифы

Green API предоставляет разные тарифы для взаимодействия с функциями сервиса:

- Бесплатный тариф **Разработчик**

Позволяет тестировать отправку и получение сообщений, делиться геопозицией, проверять, зарегистрирован ли номер клиента в WhatsApp, и множество других функций. Но на него действует ряд ограничений.

На тарифе "Разработчик" можно создать максимум 1 инстанс, взаимодействовать только с 3 чатами (контактами или группами) и получать уведомления только от 3 чатов.

- Платный тариф **Бизнес**

В нём собраны все возможности бесплатного тарифа без ограничений. Он создан для полномасштабной работы с большим количеством индивидуальных клиентов и групповых чатов.

Подробнее о тарифах, их возможностях и ограничениях:

1. [Методы и ограничения](#)
2. [Уведомления и ограничения](#)

"Примечание" Ограничения обновляются каждый месяц 1 числа.

В данных таблицах описаны методы и уведомления, на которых есть ограничения. Если метод отсутствует в таблице, то на нем нет ограничения.

Методы и ограничения

Метод	Тариф "Разработчик". Число запросов в месяц	Тариф "Business". Число запросов в месяц
checkWhatsapp	100	30 000
sendContact	1000	Безлимит
deleteMessage	1000	Безлимит
sendLocation	1000	Безлимит
sendFileByUrl	1000	Безлимит
updateGroupName	1000	Безлимит
getGroupData	1000	Безлимит
addGroupParticipant	1000	Безлимит
removeGroupParticipant	1000	Безлимит
setGroupAdmin	1000	Безлимит
removeAdmin	1000	Безлимит
leaveGroup	1000	Безлимит
sendFileByUpload	1000	Безлимит
lastIncomingMessages	1000	Безлимит
lastOutgoingMessages	1000	Безлимит
createGroup	1000	Безлимит
getContacts	1000	Безлимит
forwardMessages	1000	Безлимит
readChat	500	Безлимит
downloadFile	500	Безлимит
getChatHistory	300	Безлимит
getStatusInstance	100	Безлимит
getAvatar	100	Безлимит
setGroupPicture	100	Безлимит
archiveChat	100	Безлимит
unarchiveChat	100	Безлимит
getContactInfo	100	Безлимит
setDisappearingChat	100	Безлимит
sendTextStatus	10	Безлимит
sendVoiceStatus	10	Безлимит
sendMediaStatus	10	Безлимит
getOutgoingStatuses	10	Безлимит
getIncomingStatuses	10	Безлимит
deleteStatus	Безлимит	Безлимит

Метод	Тариф "Разработчик". Число запросов в месяц	Тариф "Business". Число запросов в месяц
getStatusStatistic	Безлимит	Безлимит
getSettings	Безлимит	Безлимит
setSettings	Безлимит	Безлимит
getStateInstance	Безлимит	Безлимит
reboot	Безлимит	Безлимит
logout	Безлимит	Безлимит
qr	Безлимит	Безлимит
scanQrCode	Безлимит	Безлимит
setProfilePicture	Безлимит	Безлимит
getWaSettings	Безлимит	Безлимит
receiveNotification	Безлимит	Безлимит
deleteNotification	Безлимит	Безлимит
getMessage	Безлимит	Безлимит
getDeviceInfo	Безлимит	Безлимит
sendMessage	Безлимит	Безлимит
sendButtons	Безлимит	Безлимит
sendTemplateButtons	Безлимит	Безлимит
sendListMessage	Безлимит	Безлимит
sendLink	Безлимит	Безлимит
sendPoll	Безлимит	Безлимит
clearMessagesQueue	Безлимит	Безлимит
sendTemplate	Безлимит	Безлимит

Уведомления и ограничения

Уведомления	Тариф "Разработчик". Число запросов в месяц	Тариф "Бизнес". Число запросов в месяц
incomingMessageReceived	Безлимит	Безлимит
outgoingMessageReceived	Безлимит	Безлимит
outgoingAPIMessageReceived	Безлимит	Безлимит
outgoingMessageStatus	Безлимит	Безлимит
stateInstanceChanged	Безлимит	Безлимит
statusInstanceChanged	Безлимит	Безлимит
deviceInfo	Уведомление временно не работает. Безлимит	Уведомление временно не работает. Безлимит
incomingCall	Безлимит	Безлимит
incomingBlock	Уведомление временно не работает. Безлимит	Уведомление временно не работает. Безлимит
quotaExceeded	Безлимит	Безлимит

1.2.3 Выполнение запросов

Для выполнения запроса к Green API необходимо выполнить GET, POST или DELETE запрос на адрес

```
{{apiUrl}}/waInstance{{idInstance}}/{{method}}/{{apiTokenInstance}}
```

Адрес запроса формируется из хоста API `apiUrl/mediaUrl`, номера инстанса `idInstance` и ключа доступа `apiTokenInstance`, которые требуется предварительно получить в [Личном кабинете](#) согласно разделу [Перед началом работы](#).

В качестве параметра `method` используется выбранный метод API, описанный в [Документации API](#).

Пример запроса для отправки сообщения методом [SendMessage](#):

```
https://api.green-api.com/waInstance1234/SendMessage/bde035edae3fc00bc116bd112297908d8145e5ba8decc5d884
```

Заголовки запроса

Большинство запросов должны содержать стандартные заголовки:

- Content-Type - application/json

В некоторых случаях заголовки могут отличаться, например, для метода отправки файла [SendFileByUpload](#)

Отладка запросов

Для отладки запросов к Green API рекомендуется использовать [Коллекцию Postman](#).

1.2.4 Коллекция Postman

Для тестирования и отладки запросов к Green API, Вы можете использовать [раздел API в личном кабинете](#) или [коллекцию Postman](#).

[Postman](#) - популярный инструмент для тестирования и разработки API. Чтобы упростить разработчикам интеграцию с Green API, мы создали [коллекцию Postman](#) с полным набором необходимых API.

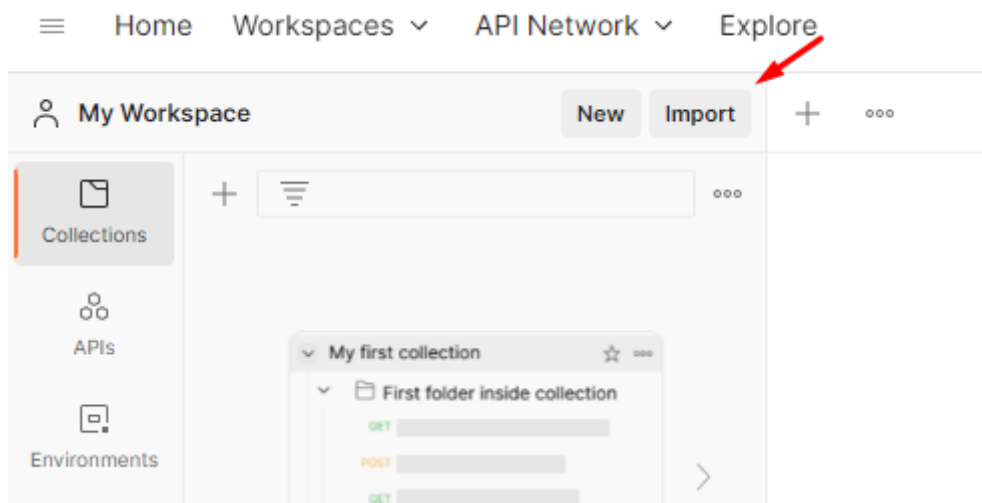
1. [Установка](#): установка коллекции
2. [Конфигурирование](#): настройка переменных среды
3. [Тестирование](#): использование методов API

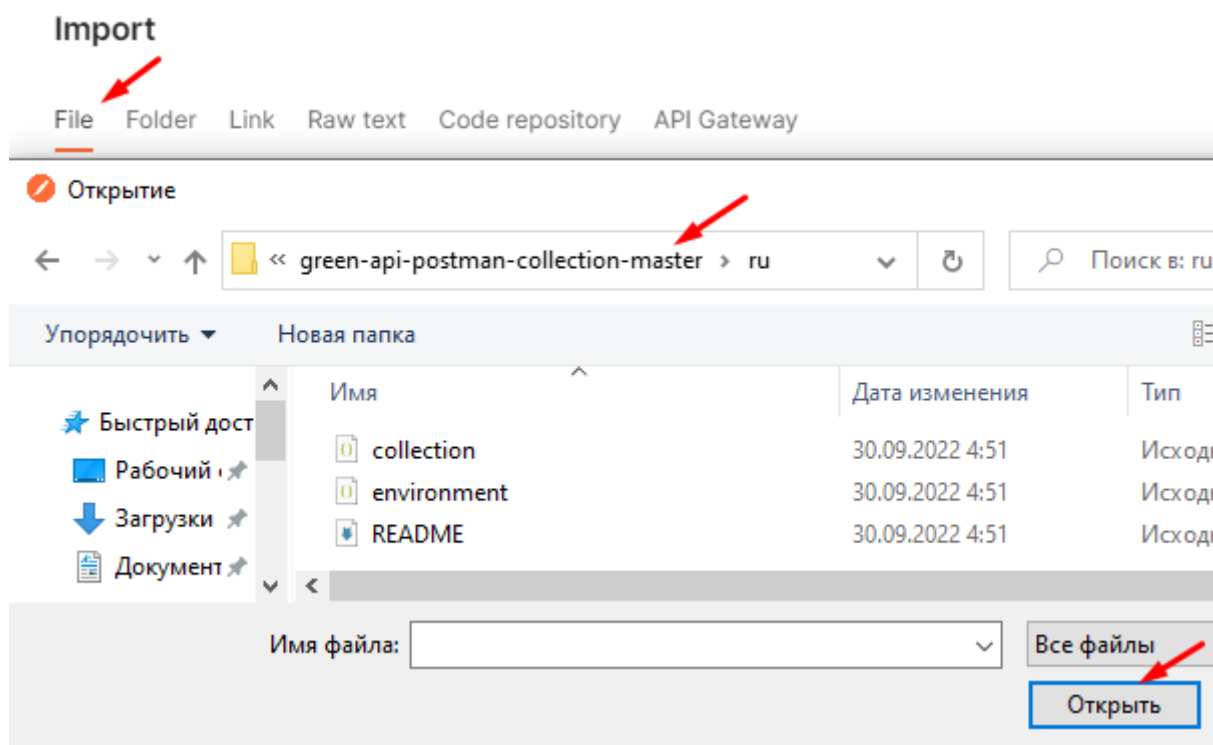
Установка

Чтобы приступить к работе, скачайте указанные ниже компоненты и установите Postman:

- Приложение [Postman](#)
- Коллекция [Green API - Postman Collection](#) (клонировать репозиторий или скачать пакет в виде ZIP-файла)

После установки и запуска Postman, нажмите `Import` и выберите два JSON-файла `collection.json` и `environment.json` из пакета коллекции [Postman на GitHub](#).





После импорта вы увидите элемент `Green API` в разделе `Collections` и сможете выбрать пункт `Green API Developer` в качестве `Environment`.

Конфигурирование

Настраиваемая среда Postman фактически представляет собой набор пар "ключ-значение". Вы можете создавать стандартные переменные, которые затем будут использоваться в разных запросах. [Подробнее о переменных среды Postman.](#)

Заранее настроенная среда `Green API Developer` содержит полный набор переменных, на которые ссылается коллекция. Некоторые из этих переменных следует отредактировать и заменить собственными значениями. Чтобы открыть диалог редактирования, нажмите маленькую кнопку с изображением глаза рядом с раскрывающимся списком среды и выберите `Edit`.

Задайте значения четырех переменных `apiUrl`, `mediaUrl`, `idInstance` и `apiTokenInstance`, которые были получены на этапе [Перед началом работы](#).

Тестирование

Теперь вы можете выбрать любой метод API в коллекции и приступить к отправке запросов. Все методы для удобства приведены в том же порядке, в каком они рассматриваются в [Документации API](#). Вы можете вносить в эти методы любые изменения, чтобы упростить их тестирование и обработку ответов.

1.3 Документация API

1.3.1 Документация API

Green API предоставляет HTTP API WhatsApp для отправки и получения сообщений, файлов, работы с групповыми чатами, получения списка контактов и других методов.

Перед выполнением запросов убедитесь, что выполнены все шаги раздела [Перед началом работы](#). Ознакомьтесь с разделом [Выполнение запросов](#). Для отладки запросов к Green API воспользуйтесь [Коллекцией Postman](#).

Аккаунт

- [Получить настройки инстанса](#)
- [Установить настройки инстанса](#)
- [Получить состояние инстанса](#)
- [Перезапустить инстанс](#)
- [Разлогинить инстанс](#)
- [Получить QR-код](#)
- [Получить QR-код через websocket](#)
- [Связать по номеру телефона](#)
- [Получить информацию об аккаунте WhatsApp](#)

Отправка

- [Отправить текст](#)
- [Отправить опрос](#)
- [Отправить видео, аудио, изображение, документ](#)
- [Отправить видео, аудио, изображение, документ по URL](#)
- [Отправить геолокацию](#)
- [Отправить контакт](#)
- [Переслать сообщения](#)

Получение

ПОЛУЧЕНИЕ УВЕДОМЛЕНИЙ ЧЕРЕЗ HTTP API

- [Получить уведомление](#)
- [Удалить уведомление](#)

ПОЛУЧЕНИЕ УВЕДОМЛЕНИЙ ЧЕРЕЗ WEBHOOK ENDPOINT

ФОРМАТ ВХОДЯЩИХ УВЕДОМЛЕНИЙ

Входящее сообщение

- [Входящее текстовое сообщение](#)
- [Выбор обычной кнопки](#)
- [Выбор шаблонной кнопки](#)
- [Выбор элемента списка](#)

- Входящее текстовое сообщение, сообщение с URL, рекламное сообщение
- Входящее сообщение с изображением, видео, аудио, документом
- Входящее сообщение с геолокацией
- Входящее сообщение с контактом
- Входящее сообщение с массивом контактов
- Входящее сообщение со стикером
- Входящее сообщение-реакция
- Входящее сообщение с приглашением в группу
- Входящее сообщение с опросом
- Входящее сообщение с обновлением опроса

Отправленное сообщение

- Отправленное с телефона сообщение
- Отправленное сообщение через API
- Статус отправленного сообщения

Прочие

- Статус инстанса
- Входящий звонок

Объекты

- Типы входящих уведомлений

ПОЛУЧЕНИЕ ФАЙЛОВ

- Скачать файл из входящего сообщения

Журналы

- Получить историю сообщений чата
- Получить сообщение чата
- Получить журнал входящих сообщений
- Получить журнал отправленных сообщений

Очереди

- Получить очередь сообщений к отправке
- Очистить очередь сообщений к отправке

Группы

- Создать группу
- Изменить имя группы
- Получить информацию о группе
- Добавить участника в группу
- Удалить участника из группы
- Назначить права администратора группы
- Отозвать права администратора группы
- Выйти из группы

Статусы (бета-версия)

СТАТУСЫ

- Отправить текстовый статус
- Отправить голосовой статус
- Отправить медиа статус
- Удалить статус

СТАТИСТИКА

- Получить статистику статуса

ИСТОРИЯ

- Получить входящие статусы
- Получить исходящие статусы

Отметка прочтения

- Отметить чат прочитанным

Сервисные методы

- Проверить наличие WhatsApp
- Получить аватар
- Получить контакты
- Получить информацию о контакте
- Удалить сообщение
- Архивировать чат
- Разархивировать чат
- Изменить настройки исчезающих сообщений чата

Прочее

- Идентификатор чата
- Интервал отправки сообщений
- Стандартные ошибки
- Достижение лимитов на тарифе Разработчик

Рекомендации по интеграции

- Создание и настройка инстанса
- Отслеживание состояния инстанса
- Подключение WhatsApp к Green API
- Работа с входящими звонками
- Использование хостов Green API
- Работа с входящими уведомлениями

1.3.2 Аккаунт

Аккаунт

В данном разделе представлены методы по работе с Аккаунтом

- [Получить настройки инстанса](#)
- [Установить настройки инстанса](#)
- [Получить состояние инстанса](#)
- [Перезапустить инстанс](#)
- [Разлогинить инстанс](#)
- [Получить QR-код](#)
- [Получить QR-код через websocket](#)
- [Связать по номеру телефона](#)
- [Установка аватара аккаунта](#)
- [Получить информацию об аккаунте](#)

GetSettings



Метод предназначен для получения текущих настроек инстанса.

ЗАПРОС

Для получения текущих настроек инстанса требуется выполнить запрос по адресу:

```
GET {{apiUrl}}/waInstance{{idInstance}}/getSettings/{{apiTokenInstance}}
```

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

ОТВЕТ

Поля ответа

Поле	Тип	Описание
wid	string	Идентификатор аккаунта в WhatsApp
countryInstance	string	Не используется
typeAccount	string	Не используется
webhookUrl	string	URL для получения входящих уведомлений. При получении уведомлений технологией HTTP API , поле должно быть пустым. Описание работы поля.
webhookUrlToken	string	Токен для доступа к вашему серверу уведомлений. Описание работы поля.
delaySendMessagesMilliseconds	integer	Интервал отправки сообщений в миллисекундах
markIncomingMessagesReaded	string	Отмечать входящие сообщения прочитанными или нет, возможные значения: <code>yes</code> , <code>no</code> . Игнорируется, если <code>markIncomingMessagesReadedOnReply</code> в значении 'yes'.
markIncomingMessagesReadedOnReply	string	Отмечать входящие сообщения прочитанными при отправке сообщения в чат через API, возможные значения: <code>yes</code> , <code>no</code> . Если в значении 'yes', то настройка <code>markIncomingMessagesReaded</code> игнорируется.
sharedSession	string	Не используется
outgoingWebhook	string	Получать уведомления о статусах отправки/доставки/прочтении исходящих сообщений, возможные значения: <code>yes</code> , <code>no</code> .
outgoingMessageWebhook	string	Получать уведомления о сообщениях, отправленных с телефона, возможные значения: <code>yes</code> , <code>no</code>
outgoingAPIMessageWebhook	string	Получать уведомления о сообщениях, отправленных через API, возможные значения: <code>yes</code> , <code>no</code> . При отправке сообщения на несуществующий аккаунт WhatsApp, уведомление не придет.
incomingWebhook	string	Получать уведомления о входящих сообщениях и файлах, возможные значения: <code>yes</code> , <code>no</code>
deviceWebhook	string	Временно не работает. Получать уведомления об устройстве (телефоне) и уровне заряда батареи, возможные значения: <code>yes</code> , <code>no</code>
statusInstanceWebhook	string	Не используется
stateWebhook	string	Получать уведомления об изменении состояния авторизации инстанса, возможные значения: <code>yes</code> , <code>no</code>
enableMessagesHistory	string	Не используется
keepOnlineStatus	string	Выставляет статус 'В сети' для вашего аккаунта
pollMessageWebhook	string	Получать уведомления о создании опроса и голосовании в опросе, возможные значения: <code>yes</code> , <code>no</code>
incomingBlockWebhook	string	Временно не работает. Получать уведомления о добавлении чата в список заблокированных контактов, возможные значения: <code>yes</code> , <code>no</code>
incomingCallWebhook	string	Получать уведомления о статусах входящего звонка, возможные значения: <code>yes</code> , <code>no</code>

"Применение настроек" Для получения уведомлений о **статусах отправки/доставки/прочтении сообщений**, отправленных с телефона требуется включить настройки:

```
* `outgoingMessageWebhook`
* `outgoingWebhook`
```

Для получения уведомлений о **статусах входящего звонка**, требуется включить настройки:

```
* `incomingCallWebhook`
* `incomingWebhook`
```

Для получения уведомлений об **опросах и об обновлениях опросов**, требуется включить настройки:

```
* Если опрос отправлен через API: `outgoingAPIMessageWebhook` и `pollMessageWebhook`
```

```
* Если опрос отправлен с телефона: `outgoingMessageWebhook` и `pollMessageWebhook`
```

```
* Для получения уведомлений о входящих опросах и об ответах на опросы: `incomingWebhook` и `pollMessageWebhook`
```

Если **телефон и связанные устройства выключены**, требуется включить настройку:

```
* `keepOnlineStatus`, чтобы выставлял статус `В сети` для вашего аккаунта и отправлял статус сообщения `delivered`
```

Пример тела ответа

```
{
  "wid": "11001234567@cs.us",
  "countryInstance": "",
  "typeAccount": "",
  "webhookUrl": "https://mysite.com/webhook/green-api/",
  "webhookUrlToken": "",
  "delaySendMessagesMilliseconds": 5000,
  "markIncomingMessagesReaded": "no",
  "markIncomingMessagesReadedOnReply": "no",
  "sharedSession": "no",
  "outgoingWebhook": "yes",
  "outgoingMessageWebhook": "yes",
  "outgoingAPIMessageWebhook": "yes",
  "incomingWebhook": "yes",
  "deviceWebhook": "no", //Уведомление временно не работает
  "statusInstanceWebhook": "no",
  "stateWebhook": "no",
  "enableMessagesHistory": "no",
  "keepOnlineStatus": "no",
  "pollMessageWebhook": "no",
  "incomingBlockWebhook": "yes", //Уведомление временно не работает
  "incomingCallWebhook": "yes"
}
```

Ошибки GetSettings

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

SetSettings



Метод предназначен для установки настроек инстанса.

При вызове данного метода инстанс перезапускается.

Настройки применяются в течение 5 минут после вызова метода `setSettings`.

После создания инстанса все настройки по умолчанию выключены.

ЗАПРОС

Для установки настроек инстанса требуется выполнить запрос по адресу:

```
POST {{apiUrl}}/waInstance/{{idInstance}}/setSettings/{{apiTokenInstance}}
```

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Допускается указывать параметры выборочно. Хотя бы один параметр должен быть указан.

Параметр	Тип	Обязательный	Описание
webhookUrl	string	Нет	URL для отправки уведомлений. Если требуется отключить получение уведомлений, то укажите пустую строку. При получении уведомлений технологией HTTP API , поле должно быть пустым. Описание работы поля
webhookUrlToken	string	Нет	Токен для доступа к вашему серверу уведомлений, если не требуется, то укажите пустую строку. Описание работы поля .
delaySendMessagesMilliseconds	integer	Нет	Интервал отправки сообщений в миллисекундах. Минимальное значение 500 мсек. Максимальное значение 86400000 мсек
markIncomingMessagesReaded	string	Нет	Отмечать входящие сообщения прочитанными или нет, возможные значения: yes, no. Игнорируется, если markIncomingMessagesReadedOnReply в значении 'yes'.
markIncomingMessagesReadedOnReply	string	Нет	Отмечать входящие сообщения прочитанными при отправке сообщения в чат через API, возможные значения: yes, no. Если в значении 'yes', то настройка markIncomingMessagesReaded игнорируется.
sharedSession	string	Нет	Не используется
countryInstance	string	Нет	Не используется
outgoingWebhook	string	Нет	Получать уведомления о статусах отправки/доставки/прочтении исходящих сообщений, возможные значения: yes, no. Статусы noAccount и failed отключить невозможно, необходимо обязательно реализовать обработку данного уведомления.
outgoingMessageWebhook	string	Нет	Получать уведомления о сообщениях, отправленных с телефона, возможные значения: yes, no
outgoingAPIMessageWebhook	string	Нет	Получать уведомления о сообщениях, отправленных через API, возможные значения: yes, no. При отправке сообщения на несуществующий аккаунт WhatsApp, уведомление не придет.
stateWebhook	string	Нет	Получать уведомления об изменении состояния авторизации инстанса, возможные значения: yes, no
incomingWebhook	string	Нет	Получать уведомления о входящих сообщениях и файлах, возможные значения: yes, no
deviceWebhook	string	Нет	

Параметр	Тип	Обязательный	Описание
			Временно не работает. Получать уведомления об устройстве (телефоне) и уровне заряда батареи, возможные значения: yes, no
statusInstanceWebhook	string	Нет	Не используется
enableMessagesHistory	string	Нет	Не используется
keepOnlineStatus	string	Нет	Выставляет статус 'В сети' для вашего аккаунта
pollMessageWebhook	string	Нет	Получать уведомления о создании опроса и голосовании в опросе, возможные значения: yes, no
incomingBlockWebhook	string	Нет	Временно не работает. Получать уведомления о добавлении чата в список заблокированных контактов, возможные значения: yes, no
incomingCallWebhook	string	Нет	Получать уведомления о статусах входящего звонка, возможные значения: yes, no

"Применение настроек" Для получения уведомлений о статусах отправки/доставки/прочтении сообщений, отправленных с телефона требуется включить настройки:

```
* `outgoingMessageWebhook`
* `outgoingWebhook`
```

Для получения уведомлений о статусах входящего звонка, требуется включить настройки:

```
* `incomingCallWebhook`
* `incomingWebhook`
```

Настройка `pollMessageWebhook` активируется только при включении следующих параметров:

```
* `outgoingAPIMessageWebhook` для сообщений, отправленных через API
* `outgoingMessageWebhook` для сообщений, отправленных с телефона
* `incomingWebhook` для входящих сообщений
```

Если телефон и связанные устройства выключены, требуется включить настройку:

```
* `keepOnlineStatus`, чтобы выставляет статус 'В сети' для вашего аккаунта и отправлять статус сообщения `delivered`
```

Пример тела запроса общий

```
{
  "webhookUrl": "https://mysite.com/webhook/green-api/",
  "webhookUrlToken": "",
  "delaySendMessagesMilliseconds": 5000,
  "markIncomingMessagesReaded": "no",
  "markIncomingMessagesReadedOnReply": "no",
  "outgoingWebhook": "yes",
  "outgoingMessageWebhook": "yes",
  "outgoingAPIMessageWebhook": "yes",
  "incomingWebhook": "yes",
  "deviceWebhook": "no", // Уведомление временно не работает
  "stateWebhook": "yes",
  "keepOnlineStatus": "no",
  "pollMessageWebhook": "no",
  "incomingBlockWebhook": "yes", // Уведомление временно не работает
  "incomingCallWebhook": "yes"
}
```

ОТВЕТ

Поля ответа

Поле	Тип	Описание
saveSettings	boolean	Флаг, что настройки сохранены

Пример тела ответа

```
{  
  "saveSettings": true  
}
```

Ошибки SetSettings

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

GetStateInstance



Метод предназначен для получения состояния инстанса.

ЗАПРОС

Для получения состояния инстанса требуется выполнить запрос по адресу:

```
GET {{apiUrl}}/waInstance{{idInstance}}/getStateInstance/{{apiTokenInstance}}
```

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

ОТВЕТ

Поля ответа

Поле	Тип	Описание
<code>stateInstance</code>	string	Состояние инстанса. Принимает значения:
		<code>notAuthorized</code> - Инстанс не авторизован. Для авторизации инстанса обратитесь к разделу Перед началом работы
		<code>authorized</code> - Инстанс авторизован
		<code>blocked</code> - Инстанс забанен
		<code>sleepMode</code> - Статус устарел. Инстанс ушел в спящий режим. Состояние возможно, когда выключен телефон. После включения телефона может потребоваться до 5 минут для перехода состояния инстанса в значение <code>authorized</code>
		<code>starting</code> - Инстанс в процессе запуска (сервисный режим). Происходит перезагрузка инстанса, сервера или инстанс в режиме обслуживания. Может потребоваться до 5 минут для перехода состояния инстанса в значение <code>authorized</code>
		<code>yellowCard</code> - На инстансе частично или полностью приостановлена отправка сообщений из-за спамерской активности. Сообщения отправленные после получения статуса хранятся в очереди к отправке 24 часа. Для продолжения работы инстанса требуется сделать перезагрузку инстанса

Пример тела ответа

```
{
  "stateInstance": "authorized"
}
```

Ошибки GetStateInstance

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
200	Инстанс долго (более 5 минут) в состоянии <code>starting</code>	<ol style="list-style-type: none"> Сделать перезапуск инстанса методом reboot Обратиться в службу технической поддержки

Reboot

Метод предназначен для перезапуска инстанса.

ЗАПРОС

Для перезапуска инстанса требуется выполнить запрос по адресу:

GET `{{apiUrl}}/waInstance{{idInstance}}/reboot/{{apiTokenInstance}}`

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

ОТВЕТ**Поля ответа**

Поле	Тип	Описание
<code>isReboot</code>	boolean	Выполнение данного метода перезагружает инстанс

Пример тела ответа

```
{
  "isReboot": true
}
```

Ошибки Reboot

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Logout



Метод предназначен для разлогинивания инстанса.

ЗАПРОС

Для разлогинивания инстанса требуется выполнить запрос по адресу:

```
GET {{apiUrl}}/waInstance{{idInstance}}/logout/{{apiTokenInstance}}
```

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

ОТВЕТ

Поля ответа

Поле	Тип	Описание
<code>isLogout</code>	boolean	Результат разлогинивания инстанса

Пример тела ответа

```
{  
  "isLogout": true  
}
```

Ошибки Logout

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

QR



Рекомендуется использовать [метод авторизации по номеру телефона](#)

Метод предназначен для получения QR-кода. Для авторизации инстанса требуется считать QR-код из приложения [WhatsApp Business](#) на телефоне. Также получить QR-код и авторизовать инстанс можно в личном кабинете. Процедура авторизации инстанса через личный кабинет описана в разделе [Перед началом работы](#).

QR-код обновляется каждые 20 секунд., поэтому вызывать метод получения QR-кода рекомендуется с интервалом в 1 секунду.

Для получения QR-кода инстанс должен быть в неавторизованном состоянии. Если инстанс авторизован, то предварительно требуется разлогинить инстанс методом [Logout](#).

После успешного сканирования QR-кода и авторизации инстанса формируется [входящее уведомление](#) с видом [Статус инстанса](#).

Также получить QR-код можно через [websocket-соединение](#)

ЗАПРОС

Для получения QR-кода требуется выполнить запрос по адресу:

```
GET {{apiUrl}}/waInstance{{idInstance}}/qr/{{apiTokenInstance}}
```

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

ОТВЕТ

Поля ответа

Поле	Тип	Описание
<code>type</code>	string	Тип сообщения, возможные значения <code>qrCode</code> , <code>error</code> , <code>alreadyLogged</code>
<code>message</code>	string	Содержание сообщения. Принимает различные значения в зависимости от значения поля <code>type</code>

Получено изображение QR-кода

Поле	Тип	Описание
<code>type</code>	string	<code>qrCode</code> - получено изображение QR-кода
<code>message</code>	string	Изображение QR-кода в кодировке <code>base64</code> . Для вывода в браузере нужно добавить строку <code>data:image/png;base64, {message}</code>

Возникла ошибка

Поле	Тип	Описание
<code>type</code>	string	<code>error</code> - возникла ошибка
<code>message</code>	string	Описание ошибки
		<code>Instance has auth. You need to make log out</code> - данные авторизации есть, но они не пригодны для авторизации, необходимо выполнить метод <code>logout</code> и произвести повторное сканирование QR кода

Получение QR кода может занимать время до 10 минут

Инстанс уже авторизован

Поле	Тип	Описание
type	string	<code>alreadyLogged</code> - инстанс уже авторизован. Для получения QR-кода требуется предварительно разлогинить инстанс методом Logout
message	string	Принимает значение <code>instance account already authorized</code>

ПРИМЕР ПОЛУЧЕНИЯ QR-КОДА В БРАУЗЕРЕ

```
https://qr.green-api.com/waInstance{{idInstance}}/{{apiTokenInstance}}
```

Для получения параметров запроса `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Вам необходимо заменить значения `idInstance` и `apiTokenInstance` на ваши, для получения ссылки вида:

```
https://qr.green-api.com/waInstance11015502/ccc44689b17435537c15a939d0a478b71c3bd7d7d52d312345
```

Также пример получения QR-кода в браузере можно посмотреть в файле [browserExampleQRcode](#).

Ошибки QR

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#).

Код HTTP	Идентификатор ошибки	Описание
200	OK	Инстанс уже авторизован
	При получении QR-кода по ссылкам выше, может появиться ошибка и загрузка кода будет происходить бесконечно долго	<ol style="list-style-type: none"> 1. Проверить корректность сформированной ссылки. 2. Проверить корректность данных <code>idInstance</code> и <code>apiTokenInstance</code>

Получить QR-код через websocket

Наравне с получением QR-кода методом [QR](#) существует возможность получить QR-код через websocket-соединение. Таймаут ожидания сканирования QR-кода составляет 100 секунд. За это время QR-код должен быть отсканирован. Для получения QR-кода инстанс должен быть в неавторизованном состоянии. Если инстанс авторизован, то предварительно требуется разлогинить инстанс методом [Logout](#).

После успешного сканирования QR-кода и авторизации инстанса формируется [входящее уведомление](#) с видом [Статус инстанса](#).

Для получения QR-кода требуется установить websocket-соединение по адресу:

```
WSS: {{apiUrl}}/waInstance{{idInstance}}/scanqrcode/{{apiTokenInstance}}
```

Для получения параметров websocket-соединения `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

ПРИМЕР ПОЛУЧЕНИЯ QR-КОДА ЧЕРЕЗ WEBSOCKET

Пример получения QR-кода через websocket можно посмотреть в файле [websocketExampleQRcode](#)

ОТВЕТ

Поля ответа

Поле	Тип	Описание
<code>type</code>	string	Тип сообщения, возможные значения <code>qrCode</code> , <code>error</code> , <code>accountData</code> , <code>alreadyLogged</code> , <code>timeout</code>
<code>message</code>	string	Содержание сообщения. Принимает различные значения в зависимости от значения поля <code>type</code>

Получено изображение QR-кода

Поле	Тип	Описание
<code>type</code>	string	<code>qrCode</code> - получено изображение QR-кода
<code>message</code>	string	Изображение QR-кода в кодировке <code>base64</code> . Для вывода в браузере нужно добавить строку <code>data:image/png;base64, {message}</code>

Возникла ошибка

Поле	Тип	Описание
<code>type</code>	string	<code>error</code> - возникла ошибка
<code>message</code>	string	Описание ошибки

Инстанс уже авторизован

Поле	Тип	Описание
<code>type</code>	string	<code>alreadyLogged</code> - инстанс уже авторизован. Для получения QR-кода требуется предварительно разлогинить инстанс методом Logout
<code>message</code>	string	Принимает значение <code>instance account already authorized</code>

Истек таймаут ожидания сканирования QR-кода

Поле	Тип	Описание
<code>type</code>	string	<code>timeout</code> - истекло время, в течение которого QR-код должен быть отсканирован. Таймаут ожидания сканирования QR-кода составляет 100 секунд.
<code>message</code>	string	Принимает значение <code>timeout</code>

GetAuthorizationCode

Метод предназначен для авторизации инстанса по номеру телефона.

Метод используется как альтернатива методу [QR](#).

Процесс авторизации:

1. На Вашем телефоне должно быть установлено официальное приложение [WhatsApp](#) или [WhatsApp Business](#).
2. Требуется зарегистрироваться в [личном кабинете](#) и создать инстанс.
3. В приложении WhatsApp выберите пункт меню "Связанные устройства" -> "Привязка устройства" -> "Связать по номеру телефона".
4. Вызовите метод GetAuthorizationCode из [личного кабинета](#) или через [коллекцию Postman](#), указав в теле запроса номер Вашего телефона (поле `phoneNumber`). В ответ на вызов метода GetAuthorizationCode вы получите 2 поля: `status` и `code`.
Код для авторизации по номеру телефона действует приблизительно 2.5 минуты.
5. Введите в приложении WhatsApp полученный код, авторизация прошла успешно.

Для получения кода инстанс должен быть в неавторизованном состоянии. Если инстанс авторизован, то предварительно требуется разлогинить инстанс методом [Logout](#).

Более подробно процедура авторизации описана в разделе [Перед началом работы](#).

После успешного ввода кода и авторизации инстанса формируется [входящее уведомление](#) вида [Статус инстанса](#).

ЗАПРОС

Для получения кода авторизации требуется выполнить запрос по адресу:

POST `{{apiUrl}}/waInstance{{idInstance}}/getAuthorizationCode/{{apiTokenInstance}}`

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>phoneNumber</code>	integer	Да	Номер телефона в международном формате без + и 00

Пример тела запроса

```
{
  "phoneNumber": 441234567890
}
```

ОТВЕТ**Поля ответа**

Поле	Тип	Описание
<code>status</code>	boolean	Статус получения кода, возможные значения <code>true</code> , <code>false</code> <code>true</code> - код получен успешно <code>false</code> - возникла ошибка при получении кода (повторить попытку получения кода)
<code>code</code>	string	Код авторизации

Пример тела ответа

```
{
  "status": true,
  "code": "GAPI2018"
}
```

Получение кода может занимать время до 30 секунд

Ошибки

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
400	Bad Request Validation failed	Ошибка валидации
400	Validation failed. Details: Wrong format. 'phoneNumber' must contain only digits	Номер должен содержать только цифры

SetProfilePicture

Метод предназначен для установки аватара аккаунта.

ЗАПРОС

Для установки аватара аккаунта требуется выполнить запрос по адресу:

POST `{{apiUrl}}/waInstance{{idInstance}}/setProfilePicture/{{apiTokenInstance}}`

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>file</code>	file	Да	Отправляемый файл в формате *.jpg

ОТВЕТ**Поля ответа**

Поле	Тип	Описание
<code>reason</code>	string	Причина почему аватар не был установлен
<code>urlAvatar</code>	string	url установленного аватара
<code>setProfilePicture</code>	boolean	флаг результата установки аватара

Пример тела ответа

В случае успеха, в ответ на запрос, отдается JSON строка следующего вида с HTTP статусом 200:

```
{
  "reason": null,
  "urlAvatar": "https://pps.whatsapp.net/v/t61.24*****-24/23*****_*****23704_*****77468_n.jpg?ccb=11-4&oh=*****b6ccc377d6332abad7d0bb&oe=*****",
  "setProfilePicture": true
}
```

Ошибки SetProfilePicture

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
200	<code>rate-overlimit</code>	Рекомендуется вызывать метод SetProfilePicture не чаще 1 раза в 10 секунд, так как существует ограничение со стороны серверов WhatsApp. Ограничение на количество запросов также указано в таблице
400	<code>File is empty</code>	Пустой файл
400	<code>Unsupported media type</code>	Неподдерживаемый формат файла

GetWaSettings



Метод предназначен для получения информации об аккаунте WhatsApp

ЗАПРОС

Для получения информации об аккаунте WhatsApp требуется выполнить запрос по адресу:

```
GET {{apiUrl}}/waInstance{{idInstance}}/getWaSettings/{{apiTokenInstance}}
```

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

ОТВЕТ

Поля ответа

Поле	Тип	Описание
<code>avatar</code>	string	Ссылка на аватар аккаунта Whatsapp
<code>phone</code>	string	Номер аккаунта Whatsapp
<code>stateInstance</code>	string	Состояние инстанса. Принимает значения: <ul style="list-style-type: none"> <code>notAuthorized</code> - Инстанс не авторизован. Для авторизации инстанса обратитесь к разделу Перед началом работы <code>authorized</code> - Инстанс авторизован <code>blocked</code> - Инстанс забанен <code>sleepMode</code> - Статус устарел. Инстанс ушел в спящий режим. Состояние возможно, когда выключен телефон. После включения телефона может потребоваться до 5 минут для перевода состояния инстанса в значение <code>authorized</code>. <code>starting</code> - Инстанс в процессе запуска (сервисный режим). Происходит перезагрузка инстанса, сервера или инстанс в режиме обслуживания. Может потребоваться до 5 минут для перевода состояния инстанса в значение <code>authorized</code>. <code>yellowCard</code> - На инстансе частично или полностью приостановлена отправка сообщений из-за спамерской активности. Сообщения отправленные после получения статуса хранятся в очереди к отправке 24 часа. Для продолжения работы инстанса требуется сделать перезагрузку инстанса
<code>deviceId</code>	string	Идентификатор устройства

Если инстанс имеет статус `notAuthorized`, `blocked`, или `starting`, поля `avatar`, `phone`, `deviceId` будут пустыми.

Пример тела ответа

```
{
  "avatar": "https://pps.whatsapp.net/v/t61.24694-24/362901986_992130525270289_51888663873132080_n.jpg?stp=dst-jpg_s96x96&ccb=11-4&oh=01_AdQyzknkSczbj9GISm0Egba4hnYLqqtAoLaLpT5GvJ5wQ&oe=64F40542&nc_cat=106",
  "phone": "0123456789",
  "stateInstance": "authorized",
  "deviceId": "0123456789:26@s.whatsapp.net"
}
```

Ошибки GetWaSettings

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

1.3.3 Отправка

Отправка

В данном разделе представлены методы для отправки текстовых сообщений, видео, аудио, изображений, документов, геолокации, контактов и ссылок

- [Отправить текст](#)
- [Отправить опрос](#)
- [Отправить видео, аудио, изображение, документ](#)
- [Отправить видео, аудио, изображение, документ по URL](#)
- [Выгрузить файл](#)
- [Отправить геолокацию](#)
- [Отправить контакт](#)
- [Переслать сообщения](#)

SendMessage



Метод предназначен для отправки текстового сообщения в личный или групповой чат. Сообщение будет добавлено в очередь на отправку. Сообщение на отправку хранится 24 часа в очереди и будет отправлено сразу же после авторизации телефона. Скорость отправки сообщений из очереди регулирует параметр [Интервал отправки сообщений](#).

"Отправка сообщения себе"

При отправке сообщений себе на номер или в группу с одним участником, [статус сообщения](#) будет sent (одна галочка).

ЗАПРОС

Для отправки текстового сообщения требуется выполнить запрос по адресу:

POST `{{apiUrl}}/waInstance{{idInstance}}/sendMessage/{{apiTokenInstance}}`

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>chatId</code>	string	Да	Идентификатор чата
<code>message</code>	string	Да	Текст сообщения. Поддерживаются символы emoji 😊. Требуется использовать кодировку UTF-8 без BOM
<code>quotedMessageId</code>	string	Нет	Идентификатор цитируемого сообщения. При заполнении параметра сообщение отправится с цитированием указанного сообщения чата. Цитирование сообщения возможно только из того же чата, в который производится отправка. Для отправки сообщений из другого чата используется метод ForwardMessages
<code>linkPreview</code>	boolean	Нет	Параметр включает отображение превью и описание ссылки. По умолчанию включен. Принимает значения: true/false

Максимальная длина текстового сообщения составляет 20000 символов

Пример тела запроса

Отправка сообщения в личный чат:

```
{
  "chatId": "11001234567@c.us",
  "message": "I use Green-API to send this message to you!"
}
```

Отправка сообщения в групповой чат:

```
{
  "chatId": "120363043968066561@g.us",
  "message": "I use Green-API to send this message to you!"
}
```

Отправка сообщения с цитированием:

```
{
  "chatId": "11001234567@c.us",
  "message": "I use Green-API to send this message to you!",
  "quotedMessageId": "11001234567@c.us"
}
```

```
"quotedMessageId": "361B0E63F2FDF9590386A9C9A102F34B"
}
```

ОТВЕТ

Поля ответа

Поле	Тип	Описание
idMessage	string	Идентификатор отправленного сообщения

Пример тела ответа

```
{
  "idMessage": "3EB0C767D097B7C7C030"
}
```

Ошибки SendMessage

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
400	Bad Request Validation failed	Ошибка валидации
400	Validation failed. Details: 'message' length must be less than or equal to 20000 characters long	Текст сообщения должен быть меньше или равен 20000 символам
500	request entity too large	Превышение допустимой длины json (>100кб)

"Отправка с неверным идентификатором цитируемого сообщения"

В случае некорректного указания `quotedMessageId`, система вернёт код 200 и id отправляемого сообщения, но оно не будет доставлено получателю.

SendPoll



Метод предназначен для отправки сообщения с опросом в личный или групповой чат. Сообщение будет добавлено в очередь на отправку. Сообщение на отправку хранится 24 часа в очереди и будет отправлено сразу же после авторизации телефона. Скорость отправки сообщений из очереди регулирует параметр [интервал отправки сообщений](#).

Для получения [уведомлений с входящими ответами на отправленные опросы](#) необходимо включить настройки инстанса одним из представленных вариантов:

1. Через [личный кабинет](#) включить настройки `Получать уведомления о входящих сообщениях и файлах` и `Получать уведомления об опросах`
2. Через метод [SetSettings](#) включить настройки `incomingWebhook` и `pollMessageWebhook`

ЗАПРОС

Для отправки требуется выполнить запрос по адресу:

POST `{{apiUrl}}/waInstance{{idInstance}}/sendPoll/{{apiTokenInstance}}`

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>chatId</code>	string	Да	Идентификатор чата
<code>message</code>	string	Да	Текст сообщения. Поддерживаются символы emoji 😊. Максимальная длина текстового сообщения составляет 255 символов.
<code>options</code>	array	Да	Массив данных о вариантах выбора. Количество вариантов ответов в голосовании не может быть больше 12. Варианты ответа должны отличаться друг от друга минимум одним символом.
<code>multipleAnswers</code>	boolean	Нет	Разрешить несколько ответов. <code>true</code> — включено, <code>false</code> — отключено, По умолчанию: <code>false</code>
<code>quotedMessageId</code>	string	Нет	Идентификатор цитируемого сообщения. При заполнении параметра сообщение отправится с цитированием указанного сообщения чата. Цитирование сообщения возможно только из того же чата, в который производится отправка. Для отправки сообщений из другого чата используется метод ForwardMessages

Поля массива `options`

Параметр	Тип	Описание
<code>optionName</code>	string	Текст варианта выбора. Максимальная длина текста составляет 100 символов.

Пример тела запроса

"Количество вариантов ответов в опросе" Для успешной отправки опроса количество вариантов ответа должно быть не менее 2 и не более 12.

Отправка сообщения в личный чат:

```
{
  "chatId": "11001234567@c.us",
  "message": "Please choose the color:",
  "options": [
    {
      "optionName": "green"
    },
    {
      "optionName": "red"
    },
    {
      "optionName": "blue"
    }
  ],
  "multipleAnswers": false
}
```

ОТВЕТ

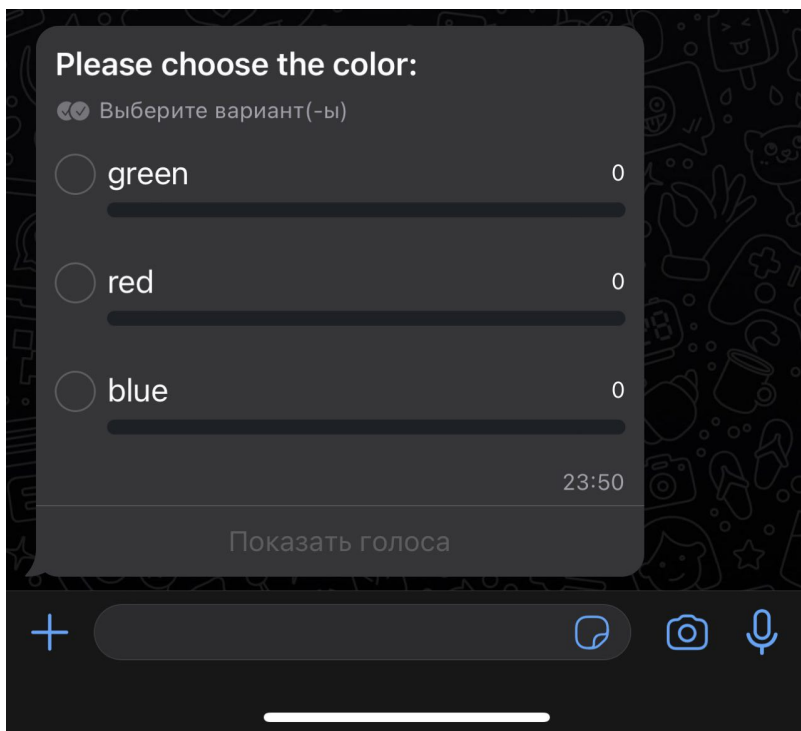
Поля ответа

Поле	Тип	Описание
idMessage	string	Идентификатор отправленного сообщения

Пример тела ответа

```
{
  "idMessage": "3E80C767D097B7C7C030"
}
```

Пример отображения у получателя



Ошибки SendPoll

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
400	Bad Request Validation failed	Ошибка валидации
400	Validation failed. Details: 'message' length must be less than or equal to 255 characters long	Длина сообщения должна быть меньше или равна 255 символам
400	Validation failed. Details: 'optionName' length must be less than or equal to 100 characters long	Длина варианта ответа должна быть меньше или равна 100 символам
400	Validation failed. Details: 'options[X].optionName' is required	Некорректный ключ для варианта ответа X
400	Validation failed. Details: 'options' length must be from 2 to 12 elements inclusive	Некорректное количество вариантов ответа
400	Validation failed. Details: 'optionName' field must have unique value	Варианты ответа не должны повторяться
500	request entity too large	Превышение допустимой длины json (>100кб)

"Отправка с неверным идентификатором цитируемого сообщения"

В случае некорректного указания 'quotedMessageId', система вернёт код 200 и id отправляемого сообщения, но оно не будет доставлено получателю.

SendFileByUpload



Для метода **SendFileByUpload** рекомендуется использовать хост **media.green-api.com**

Метод предназначен для отправки файла, загружаемого через форму (form-data). Сообщение будет добавлено в очередь на отправку, в ответе Вы получите ссылку на загруженный файл. Скорость отправки сообщений из очереди регулирует параметр [Интервал отправки сообщений](#).

Срок действия ссылки 15 дней.

Ссылку можно использовать повторно методом [SendFileByUrl](#).

Файлы видео, аудио и изображений отправляются, как и в родном WhatsApp с возможностью просмотра и прослушки. Документы отправляются так же как в родном WhatsApp. Тип отправляемого файла и способ его отправки определяется по расширению файла. Описание добавляется только к изображениям и видео.

Максимальный размер отправляемых файлов равен 100 Мбайт.

Отправка нескольких файлов одним запросом невозможна. 1 файл отправляется только 1 сообщением. Возможность отправки нескольких файлов одновременно формируется только на стороне приложения WhatsApp.

Ограничений на разрешения изображений нет. Однако при отправке изображений через API с разрешением свыше 3000x3000 пикселей, превью не формируется.

[Список файловых форматов](#), которые успешно отправляются и принимаются через API

ЗАПРОС

Для отправки файла требуется выполнить запрос по адресу:

```
POST {{mediaUrl}}/waInstance{{idInstance}}/sendFileByUpload/{{apiTokenInstance}}
```

Для получения параметров запроса `mediaUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>chatId</code>	string	Да	Идентификатор чата
<code>file</code>	file	Да	Отправляемый файл
<code>fileName</code>	string	Нет	Название файла. Должно содержать расширение файла. Требуется использовать кодировку UTF-8 без BOM. Например: <code>test.jpg</code>
<code>caption</code>	string	Нет	Описание под файлом. Описание добавляется к видео, изображениям. Максимальная длина поля 20000 символов.
<code>quotedMessageId</code>	string	Нет	Идентификатор цитируемого сообщения. При заполнении параметра сообщение отправится с цитированием указанного сообщения чата. Цитирование сообщения возможно только из того же чата, в который производится отправка. Для отправки сообщений из другого чата используется метод ForwardMessages

"Время на отправку файла"

Отправка файлов происходит в несколько этапов:

1. Получение файла
2. Выгрузка файла на сервер WhatsApp
3. Запрос на отправку файла в WhatsApp

Время на отправку файла зависит от размера файла, скорости получения файла и обработки файла WhatsApp. В зависимости от данных факторов время на отправку файла может меняться от 1 до 20 секунд.

ОТВЕТ

Поля ответа

Поле	Тип	Описание
idMessage	string	Идентификатор отправленного сообщения
urlFile	string	Ссылка на файл (срок действия ссылки 15 дней)

Пример тела ответа

```
{
  "idMessage": "3EB0C767D097B7C7C030",
  "urlFile": "https://sw-media-out.storage.yandexcloud.net/1101776123/c1aab48-c1c2-49b1-8f2d-f575a41777be.jpg"
}
```

Ошибки SendFileByUpload

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
200	failed to upload file	Превышение 100кб в теле запроса
400	failed to upload file	Отсутствует поле chatId / Присутствует лишнее поле / Указан некорректный messageId для цитаты / Отсутствует файл / Неверный тип поля / Используются не form-data / Некорректный номер
400	Bad Request Validation failed	Ошибка валидации
400	file should not be empty	Пользователь отправил пустой файл. Загружаемый файл не должен быть пустым.
413	request entity too large	Превышение допустимого размера файла (~99.9mb) Возникает при отправке файлов в обработке 1С (версия 8.3.22.1923). Возможное решение: Сменить версию 1С
500	Internal Server Error failed to upload file	Отсутствует поле file
500	Internal Server Error	Указан неверный метод для запроса. Укажите метод POST и повторите отправку

"Отправка с неверным идентификатором цитируемого сообщения"

В случае некорректного указания 'quotedMessageId', система вернёт код 200 и id отправляемого сообщения, но оно не будет доставлено получателю.

SendFileByUrl

Метод предназначен для отправки файла, загружаемого по ссылке. Сообщение будет добавлено в очередь на отправку. Сообщение на отправку хранится 24 часа в очереди и будет отправлено сразу же после авторизации телефона. Скорость отправки сообщений из очереди регулирует параметр [Интервал отправки сообщений](#).

Для формирования ссылки на файл, вы можете воспользоваться [сервисами внешнего хранения файлов](#). Мы рекомендуем использовать наше внешнее хранилище файлов с помощью метода [UploadFile](#).

Файлы видео, аудио и изображений отправляются, как и в родном WhatsApp с возможностью просмотра и прослушки. Документы отправляются так же как в родном WhatsApp. Тип отправляемого файла и способ его отправки определяется по расширению файла. Описание добавляется только к изображениям и видео.

Максимальный размер отправляемых файлов равен 100 Мбайт.

Отправка нескольких файлов одним запросом невозможна. 1 файл отправляется только 1 сообщением. Возможность отправки нескольких файлов одновременно формируется только на стороне приложения WhatsApp.

Ограничений на разрешения изображений нет. Однако при отправке изображений через API с разрешением свыше 3000x3000 пикселей, превью не формируется.

[Список файловых форматов](#), которые успешно отправляются и принимаются через API

ЗАПРОС

Для отправки файла требуется выполнить запрос по адресу:

POST

```
{{apiUrl}}/waInstance{{idInstance}}/sendFileByUrl/{{apiTokenInstance}}
```

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>chatId</code>	string	Да	Идентификатор чата
<code>urlFile</code>	string	Да	Ссылка на отправляемый файл
<code>fileName</code>	string	Да	Название файла. Должно содержать расширение файла. Требуется использовать кодировку UTF-8 без BOM. Например: <code>test.jpg</code>
<code>caption</code>	string	Нет	Описание под файлом. Описание добавляется к видео, изображениям. Максимальная длина поля 20000 символов.
<code>quotedMessageId</code>	string	Нет	Идентификатор цитируемого сообщения. При заполнении параметра сообщение отправится с цитированием указанного сообщения чата. Цитирование сообщения возможно только из того же чата, в который производится отправка. Для отправки сообщений из другого чата используется метод ForwardMessages

"Примечание"

* Рекомендации по формату ссылок

1. В ссылке могут быть использованы символы a-z, A-Z, 0-9, - (дефис), _ (нижнее подчеркивание)
2. В ссылке должны отсутствовать спец символы ~\$€#E?! и пробелы
3. Кодировка символов ссылки UTF-8 без BOM
4. Требуется передавать не закодированные ссылки (система сама кодирует ссылку)
5. Требуется чтобы ссылка указывала на конкретный файл, а не на html страницу
6. Требуется использовать быстрое хранилище, без ограничения на количество запросов файлов

* Поле fileName

Требуется обязательно указывать поле fileName с верным расширением файла.

Определение типа файла системой:

1. По расширению указанного в названии файла, в поле fileName
2. Если поле fileName не содержит расширения, то тип файла получаем по Content-Type из заголовка ссылки
3. Если в заголовке поле Content-Type отсутствует или содержит несуществующий тип файла, система постарается определить тип файла по байтам файла
4. Если система не сможет определить тип файла, то произойдет отправка с типом по умолчанию (бинарный файл)

"Время на отправку файла"

Отправка файлов происходит в несколько этапов:

1. Получение файла
2. Выгрузка файла на сервер WhatsApp
3. Запрос на отправку файла в WhatsApp

Время на отправку файла зависит от размера файла, скорости получения файла и обработки файла WhatsApp. В зависимости от данных факторов время на отправку файла может меняться от 1 до 20 секунд.

Пример тела запроса

Отправка сообщения в личный чат:

```
{
  "chatId": "11001234567@c.us",
  "urlFile": "https://my.site.com/img/horse.png",
  "fileName": "horse.png",
  "caption": "Лошадка"
}
```

Отправка сообщения в групповой чат:

```
{
  "chatId": "120363043968066561@g.us",
  "urlFile": "https://my.site.com/img/horse.png",
  "fileName": "horse.png",
  "caption": "Лошадка"
}
```

Отправка сообщения с цитированием:

```
{
  "chatId": "11001234567@c.us",
  "urlFile": "https://my.site.com/img/horse.png",
  "fileName": "horse.png",
  "caption": "Лошадка",
  "quotedMessageId": "361B0E63F2FDF95903B6A9C9A102F34B"
}
```

ОТВЕТ

Поля ответа

Поле	Тип	Описание
idMessage	string	Идентификатор отправленного сообщения

Пример тела ответа

```
{
  "idMessage": "3EB0C767D097B7C7C030"
}
```

Ошибки SendFileByUrl

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
400	Bad Request Validation failed	Ошибка валидации
400	Validation failed. Details: url has incorrect format. It should start with http(s)://'	Некорректная ссылка
500	request entity too large	Превышение допустимой длины Json (>100кб)

"Отправка с неверным идентификатором цитируемого сообщения"

В случае некорректного указания `quotedMessageId`, система вернёт код 200 и id отправляемого сообщения, но оно не будет доставлено получателю.

UploadFile



Для метода UploadFile рекомендуется использовать хост `mediaUrl` из личного кабинета

Метод предназначен для загрузки файла в облачное хранилище, который можно отправить методом `sendFileByUrl`. А так же позволяет избежать ошибок получения файлов из внешних хранилищ и ускорить отправку сообщений, например, в сценарии рассылки.

Альтернативный способ отправить файл загрузкой с диска: `sendFileByUpload`.

Рекомендуется применять метод при рассылках однотипных сообщений:

1. Методом UploadFile загрузите требуемый файл в хранилище Green-API, в ответе получите ссылку на файл.
2. Методом `SendFileByUrl` производите отправку используя полученную ссылку.

Срок жизни ссылки 15 дней.

Максимальный размер отправляемых файлов равен 100 Мбайт.

Тип отправляемого файла и способ его отправки определяется по расширению файла.

[Список файловых форматов](#), которые успешно отправляются и принимаются через API

ЗАПРОС

Для выгрузки файла требуется выполнить запрос по адресу:

POST `{{mediaUrl}}/waInstance{{idInstance}}/uploadFile/{{apiTokenInstance}}`

Для получения параметров запроса `mediaUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры заголовка

Параметр	Обязательный	Описание
Content-Type	Да	Тип файла в соответствии со списком MIME-типов . (Если тип файла отсутствует в списке, то вместо данной строки передаётся <code>GA-Filename</code>)
GA-Filename	Нет	Имя файла с расширением, если файл отсутствует в списке MIME-типов. Например, файлы как <code>.go</code> или <code>.py</code> . (Если тип файла соответствует списку , строка не передаётся, используется только <code>Content-Type</code>)

Если система не сможет определить тип файла, то произойдет отправка с типом по умолчанию (бинарный файл): `.bin`

Параметры запроса

Параметр	Тип	Обязательный	Описание
file	binary	Да	Выгружаемый файл

Пример тела запроса

В качестве тела запроса необходимо указать сам файл

ОТВЕТ

Поля ответа

Поле	Тип	Описание
urlFile	string	Ссылка на выгруженный файл

"Как получить информацию о сроке хранения файла в хранилище GREEN API"

Чтобы получить информацию о сроке хранения файла, нужно выполнить запрос типа HEAD.

Для этого:

1. Используя Postman создайте запрос типа `HEAD`
2. В заголовке ответа найдите поле `X-Amz-Expiration` с указанием даты и времени хранения файла.
3. Полученное значение указывает до какого числа и времени будет доступна ссылка.

Например, поле `X-Amz-Expiration`, содержит данные `expiry-date="Sat, 01 Jun 2024 00:00:00 GMT"`. Это обозначает, что ссылка доступна до `01.06.2024 00:00:00` по часовому поясу GMT.

Пример тела ответа

```
{
  "urlFile": "https://sw-media-out.storage.yandexcloud.net/1101123456/c1aabd48-c1c2-49b1-8f2d-f575a41777be.jpg"
}
```

Ошибки uploadFile

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
400	file should not be empty	Пользователь отправил пустой файл. Загружаемый файл не должен быть пустым.
413	request entity too large	Превышение допустимого размера файла (~99.9mb) Возникает при отправке файлов в обработке 1С (версия 8.3.22.1923). Возможное решение: Сменить версию 1С.
500	Internal Server Error	Неверно указан метод запроса. Измените метод на POST и повторите запрос

SendLocation



Метод предназначен для отправки сообщения геолокации. Сообщение будет добавлено в очередь на отправку. Сообщение на отправку хранится 24 часа в очереди и будет отправлено сразу же после авторизации телефона. Скорость отправки сообщений из очереди регулирует параметр [Интервал отправки сообщений](#).

ЗАПРОС

Для отправки сообщения геолокации требуется выполнить запрос по адресу:

POST `{{apiUrl}}/waInstance{{idInstance}}/sendLocation/{{apiTokenInstance}}`

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>chatId</code>	string	Да	Идентификатор чата
<code>nameLocation</code>	string	Нет	Название локации
<code>address</code>	string	Нет	Адрес локации
<code>latitude</code>	double	Да	Широта локации
<code>longitude</code>	double	Да	Долгота локации
<code>quotedMessageId</code>	string	Нет	Идентификатор цитируемого сообщения. При заполнении параметра сообщение отправится с цитированием указанного сообщения чата. Цитирование сообщения возможно только из того же чата, в который производится отправка. Для отправки сообщений из другого чата используется метод ForwardMessages

Пример тела запроса

Отправка сообщения в личный чат:

```
{
  "chatId": "11001234567@c.us",
  "nameLocation": "Restaurant",
  "address": "123456, Perm",
  "latitude": 12.3456789,
  "longitude": 10.1112131
}
```

Отправка сообщения в групповой чат:

```
{
  "chatId": "120363043968066561@g.us",
  "nameLocation": "Restaurant",
  "address": "123456, Perm",
  "latitude": 12.3456789,
  "longitude": 10.1112131
}
```

Отправка сообщения с цитированием:

```
{
  "chatId": "11001234567@c.us",
  "nameLocation": "Restaurant",
  "address": "123456, Perm",
  "quotedMessageId": "123456789"
}
```

```

"latitude": 12.3456789,
"longitude": 10.1112131,
"quotedMessageId": "36180E63F2FDF9590386A9C9A102F34B"
}

```

ОТВЕТ

Поля ответа

Поле	Тип	Описание
idMessage	string	Идентификатор отправленного сообщения

Пример тела ответа

```

{
  "idMessage": "3EB0C767D097B7C7C030"
}

```

Ошибки SendLocation

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
400	Bad Request Validation failed	Ошибка валидации
400	Validation failed. Details: 'latitude' must be a number or string	Неверный тип данных поля latitude
400	Validation failed. Details: 'latitude' must be a safe number	Значение поля latitude превышает максимальное безопасное целочисленное значение
500	Internal Server Error request entity too large	Превышение допустимой длины json (>100кб)

"Отправка с неверным идентификатором цитируемого сообщения"

В случае некорректного указания 'quotedMessageId', система вернёт код 200 и id отправляемого сообщения, но оно не будет доставлено получателю.

SendContact

Метод предназначен для отправки сообщения с контактом. Формируется визитная карточка контакта и отправляется в чат. Сообщение будет добавлено в очередь на отправку. Сообщение на отправку хранится 24 часа в очереди и будет отправлено сразу же после авторизации телефона. Скорость отправки сообщений из очереди регулирует параметр [Интервал отправки сообщений](#).

ЗАПРОС

Для отправки сообщения с контактом требуется выполнить запрос по адресу:

POST `{{apiUrl}}/waInstance{{idInstance}}/sendContact/{{apiTokenInstance}}`

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>chatId</code>	string	Да	Идентификатор чата
<code>contact</code>	object	Да	Объект о контакте
<code>quotedMessageId</code>	string	Нет	Идентификатор цитируемого сообщения. При заполнении параметра сообщение отправится с цитированием указанного сообщения чата. Цитирование сообщения возможно только из того же чата, в который производится отправка. Для отправки сообщений из другого чата используется метод ForwardMessages

Параметры объекта `contact` :

Параметр	Тип	Обязательный	Описание
<code>phoneContact</code>	integer	Да	номер телефона контакта в международном формате (без +) 11 или 12 цифр
<code>firstName</code>	string	Если не указаны <code>middleName</code> , <code>lastName</code> , <code>company</code>	Имя контакта
<code>middleName</code>	string	Если не указаны <code>firstName</code> , <code>lastName</code> , <code>company</code>	Отчество контакта
<code>lastName</code>	string	Если не указаны <code>middleName</code> , <code>firstName</code> , <code>company</code>	Фамилия контакта
<code>company</code>	string	Если не указаны <code>middleName</code> , <code>lastName</code> , <code>firstName</code>	Название компании контакта

Пример тела запроса

Отправка сообщения в личный чат:

```
{
  "chatId": "11001234567@c.us",
  "contact": {
    "phoneContact": 79001234568,
    "firstName": "Арте́м",
```

```

    "middleName": "Петрович",
    "lastName": "Евпаторийский",
    "company": "Велосипед"
  }
}

```

Отправка сообщения в групповой чат:

```

{
  "chatId": "120363043968066561@g.us",
  "contact": {
    "phoneContact": 79001234568,
    "firstName": "Артем",
    "middleName": "Петрович",
    "lastName": "Евпаторийский",
    "company": "Велосипед"
  }
}

```

Отправка сообщения с цитированием:

```

{
  "chatId": "11001234567@c.us",
  "quotedMessageId": "361B0E63F2FDF95903B6A9C9A102F34B",
  "contact": {
    "phoneContact": 79001234568,
    "firstName": "Артем",
    "middleName": "Петрович",
    "lastName": "Евпаторийский",
    "company": "Велосипед"
  }
}

```

ОТВЕТ

Поля ответа

Поле	Тип	Описание
idMessage	string	Идентификатор отправленного сообщения

Пример тела ответа

```

{
  "idMessage": "3EB0C767D097B7C7C030"
}

```

Ошибки SendContact

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
400	Bad Request Validation failed	Ошибка валидации
400	Validation failed. Details: 'value' must have at least 1 key	Поле <code>value</code> должно содержать минимум 1 символ
400	Validation failed. Details: 'contact.phoneContact' must be a number	Неверный тип данных поля <code>contact.phoneContact</code>
400	Validation failed. Details: 'contact.phoneContact' must be a safe number	Значение поля <code>contact.phoneContact</code> превышает максимальное безопасное целочисленное значение
500	Internal Server Error request entity too large	Превышение допустимой длины json (>100кб)

"Отправка с неверным идентификатором цитируемого сообщения"

В случае некорректного указания `quotedMessageId`, система вернёт код 200 и id отправляемого сообщения, но оно не будет доставлено получателю.

ForwardMessages

Метод предназначен для пересылки сообщений в личный или групповой чат. Пересылаемое сообщение будет добавлено в очередь на отправку. Сообщение на отправку хранится 24 часа в очереди и будет отправлено сразу же после авторизации телефона. Скорость отправки сообщений из очереди регулирует параметр [Интервал отправки сообщений](#).

ЗАПРОС

Для отправки требуется выполнить запрос по адресу:

POST `{{apiUrl}}/waInstance{{idInstance}}/forwardMessages/{{apiTokenInstance}}`

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>chatId</code>	string	Да	Идентификатор чата , куда пересылается сообщение
<code>chatIdFrom</code>	string	Да	Идентификатор чата , с которого пересылается сообщение
<code>messages</code>	array	Да	Коллекция идентификаторов пересылаемых сообщений

Пример тела запроса

Отправка сообщения в чат:

```
{
  "chatId": "11001234567@c.us",
  "chatIdFrom": "11001234567@c.us",
  "messages": [
    "BAE587FA1CECF760",
    "BAE5608BC86F2B59"
  ]
}
```

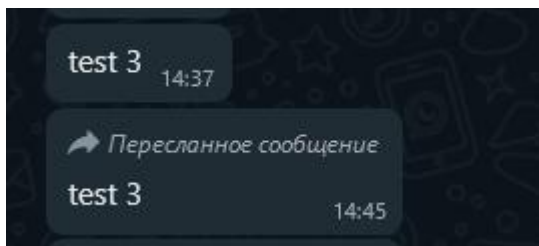
ОТВЕТ**Поля ответа**

Поле	Тип	Описание
<code>messages</code>	array	Идентификаторы отправленных сообщений

Пример тела ответа

```
{
  "messages": [
    "BAE5DBB8DEABDA22",
    "BAE5BBA9BE3142D8"
  ]
}
```

Пример отображения у получателя



Ошибки ForwardMessages

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
400	Bad Request Validation failed	Ошибка валидации
400	Validation failed. Details: 'messages' must contain at least 1 items	Ошибка валидации. Поле <code>messages</code> не должно быть пустым.

"Отправка с неверным идентификатором сообщения"

В случае некорректного указания `'messages'`, система вернёт код 200 и `id` отправляемого сообщения, но оно не будет доставлено получателю.

1.3.4 Получение

Получение данных

Получение входящих данных в Green API реализовано через входящие уведомления. На каждое событие инстанса формируется отдельное входящее уведомление. В качестве событий могут выступать: получение входящего сообщения, получение статуса ранее отправленного сообщения, входящий звонок и др. Подробнее о событиях инстанса и типах входящих уведомлений смотрите в разделе [Типы входящих уведомлений](#).

Для получения входящих уведомлений в Green API предусмотрено две технологии:

- [Получение уведомлений через HTTP API](#)
- [Получение уведомлений через Webhook Endpoint](#)

Формат уведомлений подробно описан в разделе [Формат входящих уведомлений](#)

Для скачивания файлов из входящих уведомлений обратитесь к разделу [Получение файлов](#)

HTTP API

ПОЛУЧЕНИЕ УВЕДОМЛЕНИЙ ЧЕРЕЗ HTTP API

Получать входящие уведомления (сообщения, статусы) можно через HTTP API запросы по аналогии, как реализованы остальные методы Green API. При этом гарантируется хронологический порядок следования уведомлений в той последовательности, в которой они были получены **FIFO**. Все входящие уведомления сохраняются в очереди и ожидают своего получения в течение 24 часов.

Для получения входящих уведомлений требуется выполнить последовательно вызов двух методов [ReceiveNotification](#) и [DeleteNotification](#). Метод [ReceiveNotification](#) выполняет получение входящего уведомления. Метод [DeleteNotification](#) подтверждает успешное получение и обработку уведомления. Подробнее о методах смотрите в соответствующих разделах [ReceiveNotification](#) и [DeleteNotification](#).

Срок хранения входящих уведомлений составляет 24 часа.

Настройка инстанса

Перед получением входящих уведомлений требуется выполнить настройку инстанса. Настройка инстанса может быть выполнена **программно** с использованием метода [SetSettings](#), либо **интерактивно** в интерфейсе личного кабинета.

Настройка методом [SetSettings](#)

Для настройки получения входящих уведомлений по технологии HTTP API требуется оставить пустым параметр `webhookUrl`.

Также требуется указать какие виды уведомлений необходимо получать. Для включения входящих уведомлений по видам, а также для указания параметра `webhookUrl` воспользуйтесь методом [SetSettings](#).

Пример тела запроса метода [SetSettings](#)

```
{
  "webhookUrl": "",
  "outgoingWebhook": "yes",
  "stateWebhook": "yes",
  "incomingWebhook": "yes"
}
```

Настройка в личном кабинете

Настройку получения входящих уведомлений можно также выполнить интерактивно. Для этого перейдите в [Личный кабинет](#) и выберите требуемый инстанс. Будут отображены текущее состояние инстанса и настройки для получения входящих уведомлений. Что бы поменять настройки необходимо нажать кнопку изменить на панели инстанса. Текущие настройки можно будет поменять на панели уведомлений см. рис. Укажите значение параметра `webhookUrl`, а также переключатели по видам уведомлений. Нажмите на кнопку "Сохранить изменения" для применения настроек.

Уведомления

Адрес отправки уведомлений (URL)	<input type="text" value="Введите текст"/>
Заголовок авторизации для отправки уведомлений	<input type="text" value="Введите текст"/>
Получать уведомления о входящих сообщениях и файлах	<input checked="" type="checkbox"/>
Получать уведомления о статусах отправленных сообщений	<input checked="" type="checkbox"/>
Получать уведомления о сообщениях, отправленных с телефона	<input checked="" type="checkbox"/>
Получать уведомления о сообщениях, отправленных с API	<input checked="" type="checkbox"/>
Получать уведомления об изменении состояния авторизации аккаунта	<input checked="" type="checkbox"/>
Получать уведомления об устройстве (телефоне) и уровне заряда батареи	<input type="checkbox"/>

Получение входящих уведомлений

После выполнения настройки инстанса можно приступить к получению уведомлений методами [ReceiveNotification](#) и [DeleteNotification](#). Пример кода получения уведомлений на [NodeJS](#) можно посмотреть в файле [ReceiveNotifications](#).

Подробное описание формата входящих уведомлений представлено в разделе [Формат входящих уведомлений](#).

RECEIVENOTIFICATION



Метод предназначен для получения одного входящего уведомления из очереди уведомлений.

Метод ReceiveNotification ожидает получения уведомления в течение 5 секунд. Вызов метода завершается с пустым ответом в случае достижения таймаута. Если в течение 5 секунд в очереди появляется уведомление, то вызов метода завершается, и метод возвращает полученное уведомление.

После получения и обработки входящего уведомления требуется удалить уведомление из очереди. Для этого требуется выполнить метод [DeleteNotification](#). После вызова метода [DeleteNotification](#) уведомление будет считаться принятым и обработанным и будет безвозвратно удалено из очереди. Таким образом следующий вызов метода [ReceiveNotification](#) вернет следующее уведомление из очереди в порядке поступления уведомлений в очередь.

Срок хранения входящих уведомлений в очереди составляет 24 часа.

Уведомления отдаются из очереди в порядке [FIFO](#)

Запрос

Для получения следующего входящего уведомления из очереди требуется выполнить запрос по адресу:

```
GET {{apiUrl}}/waInstance{{idInstance}}/receiveNotification/{{apiTokenInstance}}?receiveTimeout={{seconds}}
```

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры URL запроса

Параметр	Тип	Обязательный	Описание
receiveTimeout	integer	нет	Таймаут ожидания уведомления, принимает значение от 5 до 60 секунд (по умолчанию 5 секунд)

Ответ

Поля ответа

Поле	Тип	Описание
receiptId	integer	Идентификатор доставки для удаления входящего уведомления методом DeleteNotification
body	object	Входящее уведомление согласно Формату входящих уведомлений

Пример тела ответа

```
{
  "receiptId": 1234567,
  "body": {
    "typeWebhook": "incomingMessageReceived",
    "instanceData": {
      "idInstance": 1234,
      "wid": "11001234567@c.us",
      "typeInstance": "whatsapp"
    },
    "timestamp": 1588091580,
    "idMessage": "F7AEC1B7086ECD07E6E45923F5EDB825",
    "senderData": {
      "chatId": "79001234568@c.us",
      "sender": "79001234568@c.us",
      "senderName": "Иван",
      "senderContactName": "Иван Васильевич"
    },
    "messageData": {
      "typeMessage": "textMessage",
      "textMessageData": {
        "textMessage": "I use Green-API to send this message to you!"
      }
    }
  }
}
```

```

    }
  }
}

```

Ошибки ReceiveNotification

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
400	Parameter idInstance not an integer	Параметр idInstance не задан или содержит нецифровые символы
400	Parameter apiTokenInstance not define	Параметр apiTokenInstance не задан
400	Message cannot be received because custom webhook url is set. Go to cabinet, clear webhook url for instance: XXXXXXXXXX and wait for about 1 minute for another attempt	Поле webhookUrl не пустое. Очистите URL-адрес для инстанса XXXXXXXXXX и подождите около 1 минуты перед тем, как начать запрашивать уведомления.

DELETONOTIFICATION



Метод предназначен для удаления входящего уведомления из очереди уведомлений. Чтобы указать, какое уведомление следует удалить, используйте параметр `receiptId`. После получения и обработки входящего уведомления требуется удалить уведомление из очереди. Для этого требуется выполнить данный метод. После вызова метода уведомление будет считаться принятым и обработанным и будет безвозвратно удалено из очереди. Таким образом следующий вызов метода [ReceiveNotification](#) вернет следующее уведомление из очереди в порядке поступления уведомлений в очередь.

Срок хранения входящих уведомлений в очереди составляет 24 часа.

Уведомления отдаются из очереди в порядке [FIFO](#)

Запрос

Для удаления входящего уведомления из очереди требуется выполнить запрос по адресу:

DELETE

```
{{apiUrl}}/waInstance{{idInstance}}/deleteNotification/{{apiTokenInstance}}/{{receiptId}}
```

Обратите внимание, что требуется использовать запрос вида `DELETE`

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>receiptId</code>	integer	Да	Идентификатор доставки для удаления входящего уведомления, полученный методом ReceiveNotification

Ответ

Поля ответа

Поле	Тип	Описание
<code>result</code>	boolean	Результат удаления входящего уведомления: <code>true</code> - входящее уведомление успешно удалено из очереди; <code>false</code> - уведомление не удалено - возможно, когда уведомление было удалено ранее, или <code>receiptId</code> не соответствует ранее полученному значению методом ReceiveNotification

Пример тела ответа

```
{
  "result": true
}
```

Ошибки DeleteNotification

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
400	Parameter idInstance not an integer	Параметр idInstance не задан или содержит нецифровые символы
400	Parameter apiTokenInstance not define	Параметр apiTokenInstance не задан
400	Message cannot be received because custom webhook url is set. Go to cabinet, clear webhook url for instance: XXXXXXXXXXX and wait for about 1 minute for another attempt	Поле webhookUrl не пустое. Очистите URL-адрес для инстанса XXXXXXXXXXX и подождите около 1 минуты перед тем, как начать запрашивать уведомления.
500	Internal Server Error Parameter receiptId must be a Number!	Указанный receiptId не является числом
500	Internal Server Error Cannot read properties of undefined (reading 'findUnAckedMessage')	Не найдено сообщение, которое пытались удалить

Webhook Endpoint

ПОЛУЧЕНИЕ УВЕДОМЛЕНИЙ ЧЕРЕЗ WEBHOOK ENDPOINT

Технология Webhook Endpoint позволяет выполнять получение входящих уведомлений непосредственно на ваш сервер. Это означает, что сервер Green API будет выполнять вызов метода, опубликованного на стороне вашего сервера. Преимуществами данной технологии является максимально быстрое получение входящих уведомлений и высокая пропускная способность, ограниченная только скоростью обработки уведомлений на стороне вашего сервера. К недостаткам можно отнести сложность реализации.

Сервер Green API выполняет попытки доставки уведомлений с интервалом в 1 минуту. Поэтому, настраивайте свой сервер таким образом, чтобы он всегда был доступен для обработки входящих уведомлений, либо воспользуйтесь технологией [Получение уведомлений через HTTP API](#). Доставка входящих уведомлений гарантируется в течение 24 часов. Скорость отправки вебхуков составляет порядка 50 уведомлений в 1 секунду (20 мсек на 1 вебхук).

Настройка сервера

Для получения входящих уведомлений по технологии Webhook Endpoint потребуется выполнить следующие шаги:

- опубликовать IP-адрес в Интернете
- реализовать логику обработки входящих уведомлений на указанный IP-адрес
- если требуется для сервера, то установить Webhook URL Token

Публичный IP-адрес

Для получения входящих уведомлений требуется наличие публичного IP-адреса (endpoint), который будет доступен из сети Интернет. Таким образом сервер Green API сможет выполнить вызов вашего сервера по указанному адресу и передать входящее уведомление.

Наши публичные IP-адреса, с которых отправляются уведомления

Вы можете указать следующие IP адреса, с которых приходят уведомления от нас, в настройках безопасности вашего сервера:

```
46.101.109.139
51.250.12.167
51.250.84.44
51.250.95.149
89.169.137.216
158.160.49.84
165.22.93.202
167.172.162.71
```

Обработка входящих уведомлений

После получения входящего вызова на IP-адрес вашего сервера потребуется выполнить обработку полученного уведомления. Пример кода обработки входящего уведомления на [NodeJS](#) можно посмотреть в [файле](#)

Настройка инстанса

Перед получением входящих уведомлений требуется выполнить настройку инстанса. Настройка инстанса может быть выполнена [программно](#) с использованием метода [SetSettings](#), либо [интерактивно](#) в интерфейсе личного кабинета.

Настройка методом [SetSettings](#)

Для настройки получения входящих уведомлений по технологии Webhook Endpoint требуется указать в качестве параметра `webhookUrl` значение вашего IP-адреса или вашего доменного имени, и, если необходимо, `webhookUrlToken` для доступа на ваш сервер. Например:

```
https://84.211.100.201:3000/green-api/webhook/
```

Также требуется указать какие виды уведомлений необходимо получать. Для включения входящих уведомлений по видам, а также для указания параметра `webhookUrl` и `webhookUrlToken` воспользуйтесь методом [SetSettings](#).

Пример тела запроса метода [SetSettings](#)

```
{
  "webhookUrl": "https://84.211.100.201:3000/green-api/webhook/",
  "webhookUrlToken": "dscnsdiuafkascndjhsalbcvatsvcbasn23rfrfregvfdg54tds",
  "outgoingWebhook": "yes",
  "stateWebhook": "yes",
  "incomingWebhook": "yes",
  "deviceWebhook": "no"
}
```

Настройка в личном кабинете

Настройку получения входящих уведомлений можно также выполнить интерактивно. Для этого перейдите в [Личный кабинет](#) и выберите требуемый инстанс. Если инстанс авторизован, то будут отображены настройки для получения входящих уведомлений см. рис. Укажите значение параметра `webhookUrl`, а также переключатели по видам уведомлений и, если необходима авторизация на вашем вебхук сервере, Webhook URL Token. Если инстанс не авторизован и настройки уведомлений не отображаются, обратитесь к разделу [Перед началом работы](#).

Уведомления

Адрес отправки уведомлений (URL)	<input type="text" value="https://webhook.site/16b12e1f-2efd-4"/>
Заголовок авторизации для отправки уведомлений	<input type="text" value="23124scdsfscd"/>
Получать уведомления о входящих сообщениях и файлах	<input checked="" type="checkbox"/>
Получать уведомления о статусах отправленных сообщений	<input checked="" type="checkbox"/>
Получать уведомления о сообщениях, отправленных с телефона	<input checked="" type="checkbox"/>
Получать уведомления о сообщениях, отправленных с API	<input checked="" type="checkbox"/>
Получать уведомления об изменении состояния авторизации аккаунта	<input checked="" type="checkbox"/>
Получать уведомления об устройстве (телефоне) и уровне заряда батареи	<input type="checkbox"/>

"Авторизация"

Если параметр `'webhookUrlToken'` не является пустым, наш сервер будет отправлять запрос на ваш сервер с заголовком `'Authorization'`.

1. Если используется тип авторизации `**Basic**`, то в заголовке будет `'Basic <token>'`;
... - При использовании Basic авторизации требуется передавать в поле `'webhookUrlToken'` строку вида:

```
Basic <Token>
...
```

2. Если используется тип авторизации `**Bearer**`, то в заголовке будет `'Bearer <token>'`.
... - При использовании Bearer авторизации требуется передавать в поле `'webhookUrlToken'` строку вида:

```
Bearer <Token>
...
```

3. Если не указан тип авторизации, то в заголовке по умолчанию будет `'Bearer <token>'`;
... - В данном случае требуется передавать в поле `'webhookUrlToken'` строку вида:

```
Bearer <Token>
...
```

Получение входящих уведомлений

После выполнения настройки инстанса можно приступать к получению уведомлений. Примеры кода обработки входящих уведомлений:

- На [NodeJS](#) можно посмотреть [здесь](#).
- На [Python](#) можно посмотреть [здесь](#).

Отладка входящих уведомлений

Для отладки входящих уведомлений можно использовать любой бесплатный сервис в Интернет, например сервис [webhook.site](#). Сервис выдает уникальный адрес (URL), который требуется [установить](#) в качестве значения параметра `webhookUrl`.

Подробное описание формата входящих уведомлений представлено в разделе [Формат входящих уведомлений](#).

Пример запроса, который отправляет green-api

```
curl --request POST 'your-webhook-url-address' \  
--header 'Content-Type: application/json; charset=utf-8' \  
--data-raw '{"typeWebhook":"statusInstanceChanged","instanceData":{"idInstance":000001,"wid":"7123456789@c.us","typeInstance":"whatsapp"},"timestamp":1654553712,"statusInstance":"online"}'
```

Формат входящих уведомлений

ФОРМАТ ВХОДЯЩИХ УВЕДОМЛЕНИЙ

В данном разделе представлено описание формата входящих уведомлений: сообщений, файлов, статусов отправленных сообщений, статуса инстанса и устройства (телефона).

Входящее сообщение

- Входящее текстовое сообщение
- Входящее текстовое сообщение с URL
- Входящее сообщение с изображением, видео, аудио, документом
- Входящее сообщение с геолокацией
- Входящее сообщение с контактом
- Входящее сообщение с массивом контактов
- Входящее сообщение с цитированием
- Входящее сообщение со стикером
- Входящее сообщение-реакция
- Входящее сообщение с приглашением в группу
- Входящее сообщение с опросом
- Входящее сообщение с обновлением опроса

Отправленное сообщение

- Отправленное с телефона сообщение
- Отправленное сообщение через API
- Статус отправленного сообщения

Выбор кнопок

- Нажатие пользователя на обычную кнопку
- Нажатие пользователя на шаблонную кнопку
- Нажатие пользователя на значение списка выбора

Прочие

- Статус инстанса
- Аватар контакта
- Входящий звонок
- Статус доступности сокета
- Добавление чата в блэклист
- Превышены ограничения на тарифе

Объекты

- Типы входящих уведомлений

ВХОДЯЩЕЕ СООБЩЕНИЕ

Входящие сообщения

В данном разделе приводится описание общего формата входящих уведомлений с типом `incomingMessageReceived`. Описание всех типов входящих уведомлений представлено в разделе [Типы входящих уведомлений](#).

Система предусматривает получение уведомлений о входящих сообщениях следующих видов:

- [Входящее текстовое сообщение](#)
- [Входящее сообщение с кнопками](#)
- [Входящее сообщение с шаблонными кнопками](#)
- [Входящее сообщение со списком выбора](#)
- [Входящее текстовое сообщение с URL](#)
- [Входящее сообщение с изображением, видео, аудио, документом](#)
- [Входящее сообщение с геолокацией](#)
- [Входящее сообщение с контактом](#)
- [Входящее сообщение с массивом контактов](#)
- [Входящее сообщение с цитированием](#)
- [Входящее сообщение со стикером](#)
- [Входящее сообщение-реакция](#)
- [Входящее сообщение с приглашением в группу](#)
- [Входящее сообщение с опросом](#)
- [Входящее сообщение с обновлением опроса](#)

Поля уведомления `incomingMessageReceived`

Параметр	Тип	Описание
<code>typeWebhook</code>	string	Тип входящего уведомления. Для уведомления данного типа поле принимает значение <code>incomingMessageReceived</code>
<code>instanceData</code>	object	Данные об инстансе
<code>timestamp</code>	integer	Время наступления события в UNIX-формате
<code>idMessage</code>	string	Идентификатор входящего сообщения
<code>senderData</code>	object	Данные об отправителе сообщения или файла
<code>messageData</code>	object	Данные о принятом сообщении или файле

Поля объекта `instanceData`

Параметр	Тип	Описание
<code>idInstance</code>	integer	Идентификатор инстанса. Размер <code>integer</code> равен <code>int64</code> . Значения лежат в диапазоне от 1 до 10 разрядов
<code>wid</code>	string	Идентификатор аккаунта в формате WhatsApp
<code>typeInstance</code>	string	Тип мессенджера для инстанса

Поля объекта `senderData`

Параметр	Тип	Описание
<code>chatId</code>	string	Идентификатор чата, в котором получено сообщение или файл
<code>sender</code>	string	Идентификатор отправителя сообщения или файла
<code>chatName</code>	string	Имя чата
<code>senderName</code>	string	Имя отправителя
<code>senderContactName</code>	string	Имя отправителя из списка контактов текущего аккаунта

Поля объекта `messageData`

Объект `messageData` имеет разные поля в зависимости от типа входящего сообщения:

- Входящее текстовое сообщение
- Входящее текстовое сообщение с URL
- Входящее сообщение с изображением, видео, аудио, документом
- Входящее сообщение с геолокацией
- Входящее сообщение с контактом
- Входящее сообщение с массивом контактов
- Входящее сообщение с цитированием
- Входящее сообщение со стикером
- Входящее сообщение-реакция
- Входящее сообщение с приглашением в группу
- Входящее сообщение с опросом
- Входящее сообщение с обновлением опроса

Пример тела уведомления

```
{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@ec.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1588091580,
  "idMessage": "F7AEC1B7086ECDC7E6E45923F5EDB825",
  "senderData": {
    "chatId": "79001234568@ec.us",
    "sender": "79001234568@ec.us",
    "chatName": "Иван",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    // В зависимости от typeMessage = textMessage || imageMessage || videoMessage || documentMessage || audioMessage || locationMessage || contactMessage || extendedTextMessage ||
    quotedMessage || pollMessage || pollUpdateMessage
    ...
    ...
    ...
  }
}
```


Входящее текстовое сообщение

В данном разделе описывается формат входящего уведомления объекта `messageData` для входящего текстового сообщения. Для получения описания общего формата входящих уведомлений обратитесь к разделу [Входящие сообщения](#).

Для получения входящих уведомлений данного вида требуется выполнение двух условий:

```
typeWebhook = incomingMessageReceived
```

```
messageData.typeMessage = textMessage
```

"Настройка инстанса"

Чтобы получать входящие уведомления о входящих сообщениях, необходимо включить настройки одним из представленных вариантов:

1. Через личный кабинет включить настройку `Получать уведомления о входящих сообщениях и файлах`
2. Через метод `SetSettings` включить настройку ``incomingWebhook``

Уведомление Формат уведомления

Поля объекта `messageData`

Параметр	Тип	Описание
<code>typeMessage</code>	string	Тип принятого сообщения. Для сообщений данного типа поле принимает значение <code>textMessage</code>
<code>textMessageData</code>	object	Объект данных о текстовом сообщении
<code>quotedMessage</code>	object	Объект данных о цитируемом сообщении. Присутствует только, если само сообщение является цитатой

Поля объекта `textMessageData`

Параметр	Тип	Описание
<code>textMessage</code>	string	Текстовое сообщение
<code>isTemplateMessage</code>	boolean	Флаг указывает, было ли сообщение сгенерировано из шаблона

Поля объекта `quotedMessage`

Параметр	Тип	Описание
<code>stanzaId</code>	string	id цитируемого сообщения
<code>participant</code>	string	id отправителя цитируемого сообщения
<code>typeMessage</code>	string	Тип цитируемого сообщения

Остальные поля заполняются в зависимости от типа цитируемого сообщения и идентичны полям входящих сообщений описанных в разделе [Входящие сообщения](#)

Пример тела уведомления

```
{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@с.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1588091580,
  "idMessage": "F7AEC1B7086ECDC7E6E45923F5EDB825",
  "senderData": {
```

```

"chatId": "79001234568@с.us",
"sender": "79001234568@с.us",
"chatName": "Иван",
"senderName": "Иван",
"senderContactName": "Иван Васильевич"
},
"messageData": {
  "typeMessage": "textMessage",
  "textMessageData": {
    "textMessage": "I use Green-API to send this message to you!"
  }
}
}
}

```

Пример тела уведомления с цитатой текстового сообщения

```

{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@с.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1588091580,
  "idMessage": "F7AEC1B7086ECDC7E6E45923F5EDB825",
  "senderData": {
    "chatId": "79001234568@с.us",
    "sender": "79001234568@с.us",
    "chatName": "Иван",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    "typeMessage": "quotedMessage",
    "extendedTextMessageData": {
      "text": "Это",
      "stanzaId": "B7F33B88947D872F30FAA646FEDCDE2EC",
      "participant": "79001234568@с.us"
    },
    "quotedMessage": {
      "stanzaId": "9A73322488DCB7D9689A6112F2528C9D",
      "participant": "79001234568@с.us",
      "typeMessage": "textMessage",
      "textMessage": "Привет"
    }
  }
}
}

```

Пример тела уведомления текстового сообщения с цитатой сообщения изображения/видео/аудио/документ

```

{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@с.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1588091580,
  "idMessage": "F7AEC1B7086ECDC7E6E45923F5EDB825",
  "senderData": {
    "chatId": "79001234568@с.us",
    "sender": "79001234568@с.us",
    "chatName": "Иван",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    "typeMessage": "quotedMessage",
    "extendedTextMessageData": {
      "text": "Ответ",
      "stanzaId": "B4AA239D112CB2576897B3910FEDE26E",
      "participant": "79001230000@с.us"
    },
    "quotedMessage": {
      "stanzaId": "9A73322488DCB7D9689A6112F2528C9D",
      "participant": "79061230000@с.us",
      "typeMessage": "imageMessage",
      "downloadUrl": "",
      "caption": "",
      "jpegThumbnail": ""
    }
  }
}
}

```

Пример тела уведомления с цитатой сообщения контакт

```

{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,

```

```

    "wid": "11001234567@.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1588091580,
  "idMessage": "F7AEC1B7086E CDC7E6E45923F5EDB825",
  "senderData": {
    "chatId": "79001234568@.us",
    "sender": "79001234568@.us",
    "chatName": "Иван",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    "typeMessage": "quotedMessage",
    "extendedTextMessageData": {
      "text": "Ответ",
      "stanzaId": "B4AA239D112CB2576897B3910FEDE26E",
      "participant": "79001230000@.us"
    },
    "quotedMessage": {
      "stanzaId": "9A73322488DCB7D9689A6112F2528C9D",
      "participant": "79061230000@.us",
      "typeMessage": "contactMessage",
      "contact": {
        "displayName": "Green-API",
        "vcard": "BEGIN:VCARD\nVERSION:3.0\nn:Green-API\nitem1.TEL;wa id=79001230000\nitem1.X-ABLabel:Мобильный\nEND:VCARD"
      }
    }
  }
}

```

Пример тела уведомления с цитатой сообщения геопозиция

```

{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1588091580,
  "idMessage": "F7AEC1B7086E CDC7E6E45923F5EDB825",
  "senderData": {
    "chatId": "79001234568@.us",
    "sender": "79001234568@.us",
    "chatName": "Иван",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    "typeMessage": "quotedMessage",
    "extendedTextMessageData": {
      "text": "Адрес",
      "stanzaId": "CA565488E8806ED7D033E758E7463AB9",
      "participant": "79001230000@.us"
    },
    "quotedMessage": {
      "stanzaId": "9A73322488DCB7D9689A6112F2528C9D",
      "participant": "79060002233@.us",
      "typeMessage": "locationMessage",
      "location": {
        "nameLocation": "",
        "address": "",
        "jpegThumbnail": "",
        "latitude": 72.5922702,
        "longitude": 39.6645388
      }
    }
  }
}

```

Пример тела уведомления с цитатой сообщения ссылка

```

{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "7000000000@.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1658261933,
  "idMessage": "CDF9219DD08D3D84CD4E621122AFBFFD",
  "senderData": {
    "chatId": "79000000000@.us",
    "sender": "79000000000@.us",
    "chatName": "Иван",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    "typeMessage": "quotedMessage",

```

```

"extendedTextMessageData": {
  "text": "Hello",
  "stanzaId": "6FFC3BF49BEE0CF966397321C4E3D3DF",
  "participant": "7900000000@c.us"
},
"quotedMessage": {
  "stanzaId": "9A73322488DCB709689A6112F2528C9D",
  "participant": "7900000000@c.us",
  "typeMessage": "extendedTextMessage",
  "textMessage": "https://api.green-api.com/send/?phone=7000000000&text&type=phone_number&app_absent=0",
  "extendedTextMessage": {
    "description": " is free and offers simple, secure, reliable messaging and calling, available on phones all over the world.",
    "title": "Share on ",
    "previewType": "None",
    "jpegThumbnail": null
  }
}
}
}
}

```

Пример тела уведомления с ошибкой SWE001

```

{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    {
      "idInstance": 1100000001,
      "wid": "7900000000@c.us",
      "typeInstance": "whatsapp"
    }
  },
  "timestamp": 1700545580,
  "idMessage": "3AE2864C77312000000",
  "senderData": {
    {
      "chatId": "79000000001@c.us",
      "chatName": "Иван",
      "sender": "79000000001@c.us",
      "senderName": "Иван",
      "senderContactName": "Иван Васильевич"
    }
  },
  "messageData": {
    {
      "typeMessage": "textMessage",
      "textMessageData": {
        {
          "textMessage": "{{SWE001}}"
        }
      }
    }
  }
}

```

Пример тела уведомления с ошибкой SWE002

```

{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    {
      "idInstance": 1100000001,
      "wid": "7900000000@c.us",
      "typeInstance": "whatsapp"
    }
  },
  "timestamp": 1700488335,
  "idMessage": "3EB0D00923D87A3FA01BAE",
  "senderData": {
    {
      "chatId": "79000000000@c.us",
      "chatName": "Иван",
      "sender": "79000000000@c.us",
      "senderName": "Иван",
      "senderContactName": "Иван Васильевич"
    }
  },
  "messageData": {
    {
      "typeMessage": "documentMessage",
      "fileMessageData": {
        {
          "downloadUrl": "{{SWE002}}",
          "caption": "",
          "fileName": "Harry Potter and the Philosopher's Stone.pdf",
          "jpegThumbnail": "",
          "mimeType": "application/pdf",
          "forwardingScore": 0,
          "isForwarded": false
        }
      }
    }
  }
}

```

Пример тела уведомления с ошибкой SWE003

```

{
  "typeWebhook": "incomingMessageReceived",

```

```

"instanceData":
  {
    "idInstance":110000001,
    "wid":"7900000000@c.us",
    "typeInstance":"whatsapp"
  },
"timestamp":1700218861,
"idMessage":"6E112CC05BA05DD9DF",
"senderData":
  {
    "chatId":"7900000000@c.us",
    "chatName":"Иван",
    "sender":"7900000000@c.us",
    "senderName":"Иван",
    "senderContactName": "Иван Васильевич"
  },
"messageData":
  {
    "typeMessage":"textMessage",
    "textMessageData":
      {
        "textMessage":"{{SWE003}}"
      }
  }
}

```

Пример тела уведомления с ошибкой SWE999

```

{
  "typeWebhook": "incomingMessageReceived",
  "instanceData":
    {
      "idInstance":110000001,
      "wid":"7900000000@c.us",
      "typeInstance":"whatsapp"
    },
  "timestamp":1700545580,
  "idMessage":"3AE2864C77312000000",
  "senderData":
    {
      "chatId":"79000000001@c.us",
      "chatName":"Иван",
      "sender":"79000000001@c.us",
      "senderName":"Иван",
      "senderContactName": "Иван Васильевич"
    },
  "messageData":
    {
      "typeMessage":"textMessage",
      "textMessageData":
        {
          "textMessage":"{{SWE999}}"
        }
    }
}

```

Входящее текстовое сообщение, сообщение с URL или рекламное сообщение

В данном разделе описывается формат входящего уведомления объекта `messageData` для входящего текстового сообщения или сообщения с URL. Для получения описания общего формата входящих уведомлений обратитесь к разделу [Входящие сообщения](#).

Для получения входящих уведомлений данного вида требуется выполнение двух условий:

```
typeWebhook = incomingMessageReceived
```

```
messageData.typeMessage = extendedTextMessage
```

"Настройка инстанса"

Чтобы получать входящие уведомления о входящих сообщениях с URL или рекламных сообщениях, необходимо включить настройки одним из представленных вариантов:

1. Через личный кабинет включить настройку `Получать уведомления о входящих сообщениях и файлах`
2. Через метод `SetSettings` включить настройку `incomingWebhook`

Уведомление Формат уведомления

Поля объекта `messageData`

Параметр	Тип	Описание
<code>typeMessage</code>	string	Тип принятого сообщения. Для сообщений данного типа поле принимает значение <code>extendedTextMessage</code>
<code>extendedTextMessageData</code>	object	Объект данных о принятом текстовом сообщении или URL ссылки
<code>quotedMessage</code>	object	Объект данных о цитируемом сообщении. Присутствует только, если само сообщение является цитатой

Поля объекта `extendedTextMessageData`

Параметр	Тип	Описание
<code>text</code>	string	Текст ссылки или обычный текст
<code>description</code>	string	Описание ссылки, может быть пустым
<code>title</code>	string	Заголовок ссылки, может быть пустым
<code>containsAutoReply</code>	boolean	Содержит ли сообщение автоответ (только для рекламных сообщений)
<code>mediaType</code>	string	Тип медиа сообщения, например "IMAGE" (только для рекламных сообщений)
<code>showAdAttribution</code>	boolean	Показывать ли атрибуты рекламы в сообщении (только для рекламных сообщений)
<code>sourceId</code>	string	Id рекламы (только для рекламных сообщений)
<code>sourceType</code>	string	Тип источника, например "ad" (только для рекламных сообщений)
<code>sourceUrl</code>	string	Ссылка на рекламу (только для рекламных сообщений)
<code>conversionSource</code>	string	Локализация сообщения (только для рекламных сообщений)
<code>entryPointConversionApp</code>	string	Источник сообщения (только для рекламных сообщений)
<code>jpegThumbnail</code>	string	Превью изображения в base64 кодировке, может отсутствовать
<code>thumbnailUrl</code>	string	Ссылка на превью изображения (только для рекламных сообщений)
<code>isForwarded</code>	boolean	Является ли сообщение пересланным, принимает значения true/false
<code>forwardingScore</code>	integer	Количество пересылок сообщения
<code>previewType</code>	string	Тип превью (только для рекламных сообщений) Значение <code>previewType: None</code> устаревшее

Поля объекта `quotedMessage`

Параметр	Тип	Описание
<code>stanzaId</code>	string	id цитируемого сообщения
<code>participant</code>	string	id отправителя цитируемого сообщения
<code>typeMessage</code>	string	Тип цитируемого сообщения

Остальные поля заполняются в зависимости от типа цитируемого сообщения и идентичны полям входящих сообщений описанных в разделе [Входящие сообщения](#)

Пример тела уведомления

```
{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@ec.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1588091580,
  "idMessage": "F7AEC1B7086ECCDC7E6E45923F5EDB825",
  "senderData": {
    "chatId": "79001234568@ec.us",
    "sender": "79001234568@ec.us",
    "chatName": "Иван",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    "typeMessage": "extendedTextMessage",
```

```

    "extendedTextMessageData": {
      "text": "https://green-api.com/docs/video",
      "description": "Green API docs shows how you can develop the WhatsApp Bot",
      "title": "How to develop WhatsApp Bot",
      "jpegThumbnail": "/9j/4AAQSkZJRgABAQAAQABAAD/2wBDAAIEBQYFBAQYBQYHBMwYChAKGkKJChQODwwQF-xQYg==",
      "forwardingScore": 4,
      "isForwarded": true
    }
  }
}

```

Пример тела уведомления входящего сообщения со ссылкой или текстом и цитатой контакта

```

{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@c.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1588091580,
  "idMessage": "F7AEC1B7086ECCDC7E6E45923F5EDB825",
  "senderData": {
    "chatId": "79001234568@c.us",
    "sender": "79001234568@c.us",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    "typeMessage": "quotedMessage",
    "extendedTextMessageData": {
      "text": "https://yandex.ru/former&utm_source=home&utm_content=main_informer&utm_term=main_number",
      "stanzaId": "0EA554E587820E35309858AE265BE7EA",
      "participant": "79001230000@c.us",
      "forwardingScore": 4,
      "isForwarded": true
    },
    "quotedMessage": {
      "stanzaId": "9A73322488DCB7D9689A6112F2528C9D",
      "participant": "79061230000@c.us",
      "typeMessage": "contactMessage",
      "contact": {
        "displayName": "Green-API",
        "vcard": "BEGIN:VCARD\nVERSION:3.0\nn:Green-API\nitem1.TEL;waid=79001230000\nitem1.X-ABLabel:Мобильный\nEND:VCARD"
      }
    }
  }
}

```

Пример тела уведомления входящего сообщения со ссылкой и цитатой изображения

```

{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@c.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1588091580,
  "idMessage": "F7AEC1B7086ECCDC7E6E45923F5EDB825",
  "senderData": {
    "chatId": "79001234568@c.us",
    "sender": "79001234568@c.us",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    "typeMessage": "quotedMessage",
    "extendedTextMessageData": {
      "text": "https://yandex.ru/pogoda/?utm_medium=source&utm_content=main_informer&utm_term=main_number",
      "stanzaId": "B4AA239D112CB2576897B3910FEDE26E",
      "participant": "79001230000@c.us",
      "forwardingScore": 4,
      "isForwarded": true
    },
    "quotedMessage": {
      "stanzaId": "9A73322488DCB7D9689A6112F2528C9D",
      "participant": "79061230000@c.us",
      "typeMessage": "imageMessage",
      "downloadUrl": "",
      "caption": "",
      "jpegThumbnail": ""
    }
  }
}

```

Пример тела уведомления входящего рекламного сообщения

```

{
  "typeWebhook": "incomingMessageReceived",

```



```

"instanceData": {
  "idInstance": 1234,
  "wid": "11001234567@cs.us",
  "typeInstance": "whatsapp"
},
"timestamp": 1588091580,
"idMessage": "F7AEC1B7086ECCDC7E6E45923F5E0B825",
"senderData": {
  "chatId": "79001234568@cs.us",
  "chatName": "Иван",
  "sender": "79001234568@cs.us",
  "senderName": "Иван",
  "senderContactName": "Иван Васильевич"
},
"messageData": {
  "typeMessage": "extendedTextMessage",
  "extendedTextMessageData": {
    "text": "https://fb.me/gUdvBdhth",
    "description": "Новый инстанс для вашего бизнеса!",
    "title": "Green API",
    "containsAutoReply": false,
    "mediaType": "IMAGE",
    "showAdAttribution": false,
    "sourceId": "122210486228016873",
    "sourceType": "post",
    "sourceUrl": "https://fb.me/gUdvBdhth",
    "conversionSource": "FB_Ads",
    "entryPointConversionApp": "instagram",
    "jpegThumbnail": "/9j/4AAQSkZJRgABAQAAQABAAD/7QCEUghvdG9zaG9wIDMuMAA4qIJuUeMEjYwGc9BtFMXTYvgsnm4ADy6HLLZDqn1Ye5RMev+ae1hED2B9dQIARwjvDIwYcg4/r/9k=",
    "thumbnailUrl": "https://scontent.xx.fbcdn.net/v/t39.30808-6/436945887_122115998378256688_307643393252141869_n.jpg?stp=dst-jpg_s851x315&nc_cat=111&ccb=1-7&nc_sid=5f2048&nc_ohc=x61r6W-VrJIAb69T4Dl&nc_ad=z-m&nc_ci=d=0&nc_ht=scontent.xx&oh=00_AfDLFkjNOBYEntBu6SVaU0xDuxvarjKha9Ez5RQUXjsfg&oe=662D7680",
    "previewType": ""
  }
}
}
}

```

Входящее сообщение с изображением, видео, аудио, документом

В данном разделе описывается формат входящего уведомления объекта `messageData` для входящего сообщения с изображением, видео, аудио или документом. Для получения описания общего формата входящих уведомлений обратитесь к разделу [Входящие сообщения](#).

Для получения входящих уведомлений данного вида требуется выполнение двух условий:

```
typeWebhook = incomingMessageReceived
```

```
messageData.typeMessage = imageMessage || videoMessage || documentMessage || audioMessage
```

"Настройка инстанса"

Чтобы получать входящие уведомления о сообщениях с изображением, видео, аудио, документом, необходимо включить настройки одним из представленных вариантов:

1. Через личный кабинет включить настройку `Получать уведомления о входящих сообщениях и файлах`
2. Через метод `SetSettings` включить настройку `incomingWebhook``

Уведомление **Формат уведомления**

Поля объекта `messageData`

Параметр	Тип	Описание
<code>typeMessage</code>	string	Тип принятого сообщения. Для сообщений данного типа поле принимает значение: <code>imageMessage</code> , <code>videoMessage</code> , <code>documentMessage</code> , <code>audioMessage</code>
<code>fileMessageData</code>	object	Объект данных о принятом файле
<code>quotedMessage</code>	object	Объект данных о цитируемом сообщении. Присутствует только, если само сообщение является цитатой

Поля объекта `fileMessageData`

Параметр	Тип	Описание
<code>downloadUrl</code>	string	Ссылка для скачивания файла
<code>caption</code>	string	Описание под картинкой или видео
<code>fileName</code>	string	Название файла. Поле заполняется корректно только для <code>typeMessage = documentMessage</code> , в остальных случаях название файла генерируется автоматически
<code>jpegThumbnail</code>	string	Предпросмотр изображения в base64
<code>mimeType</code>	string	Тип файла, согласно классификации Media Types
<code>isForwarded</code>	boolean	Является ли сообщение пересланным, принимает значения true/false
<code>forwardingScore</code>	integer	Количество пересылок сообщения

При отправке изображений через API с разрешением свыше 3000x3000, не будет формироваться превью. Поле `jpegThumbnail` в сообщениях и получаемых уведомлениях будет пустым.

Поля объекта `quotedMessage`

Параметр	Тип	Описание
<code>stanzaId</code>	string	id цитируемого сообщения
<code>participant</code>	string	id отправителя цитируемого сообщения
<code>typeMessage</code>	string	Тип цитируемого сообщения

Остальные поля заполняются в зависимости от типа цитируемого сообщения и идентичны полям входящих сообщений описанных в разделе [Входящие сообщения](#)

Пример тела уведомления

```
{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@с.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1588091580,
  "idMessage": "F7AEC1B7086ECCDC7E6E45923F5EDB825",
  "senderData": {
    "chatId": "79001234568@с.us",
    "sender": "79001234568@с.us",
    "chatName": "Иван",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    "typeMessage": "imageMessage",
    "fileMessageData": {
      "downloadUrl": "https://api.green-api.com/waInstance1234/downloadFile/19136A974392F8C584D70DD0E1AEDF",
      "caption": "Картинка",
      "jpegThumbnail": "",
      "mimeType": "image/jpeg",
      "forwardingScore": 4,
      "isForwarded": true
    }
  }
}
```

Пример тела уведомления входящего сообщения с аудио и цитатой текстового сообщения

```
{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@с.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1588091580,
  "idMessage": "F7AEC1B7086ECCDC7E6E45923F5EDB825",
  "senderData": {
    "chatId": "79001234568@с.us",
    "sender": "79001234568@с.us",
    "chatName": "Иван",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    "typeMessage": "audioMessage",
    "fileMessageData": {
      "downloadUrl": "https://sw-media.storage.greenapi.net/9901742665/39c20293-eb8d-abdd-5fdd1b83820a.mpga",
      "fileName": "39c20293-eb8d-abdd-5fdd1b83820a.mpga",
      "mimeType": "image/jpeg",
      "forwardingScore": 4,
      "isForwarded": true
    },
    "quotedMessage": {
      "stanzaId": "9A73322488DCB7D9689A6112F2528C9D",
      "participant": "79001234569@с.us",
      "typeMessage": "textMessage",
      "textMessage": "Привет"
    }
  }
}
```

Пример тела уведомления входящего сообщения с аудио и цитатой аудио/видео/документ

```
{
  "typeWebhook": "incomingMessageReceived",
```

```

"instanceData": {
  "idInstance": 1234,
  "wid": "11001234567@cs.us",
  "typeInstance": "whatsapp"
},
"timestamp": 1588091580,
"idMessage": "F7AEC1B7086ECCDC7E6E45923F5EDB825",
"senderData": {
  "chatId": "79001234568@cs.us",
  "sender": "79001234568@cs.us",
  "chatName": "Иван",
  "senderName": "Иван",
  "senderContactName": "Иван Васильевич"
},
"messageData": {
  "typeMessage": "audioMessage",
  "fileMessageData": {
    "downloadUrl": "https://s/990173687/801078ab-3340-4e4aa5.ogg",
    "caption": "",
    "fileName": "801078ab-3340-4f78-e9978be4aa5.ogg",
    "jpegThumbnail": "",
    "mimeType": "audio/ogg",
    "forwardingScore": 4,
    "isForwarded": true
  },
  "quotedMessage": {
    "stanzaId": "9A73322488DCB7D9689A6112F2528C9D",
    "participant": "79001234568@cs.us",
    "typeMessage": "imageMessage",
    "downloadUrl": "",
    "caption": "",
    "jpegThumbnail": ""
  }
}
}

```

Пример тела уведомления входящего сообщения с изображением и цитатой геопозиция

```

{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@cs.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1588091580,
  "idMessage": "F7AEC1B7086ECCDC7E6E45923F5EDB825",
  "senderData": {
    "chatId": "79001234568@cs.us",
    "sender": "79001234568@cs.us",
    "chatName": "Иван",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    "typeMessage": "imageMessage",
    "fileMessageData": {
      "downloadUrl": "https://sw-media.storage.greenapi.net/0b-9784-483b-8426-e8d871d6de9f.jpeg",
      "caption": "",
      "fileName": "d417740b483b-8426-e8d871d6de9f.jpeg",
      "jpegThumbnail": "",
      "mimeType": "image/jpeg",
      "forwardingScore": 4,
      "isForwarded": true
    },
    "quotedMessage": {
      "stanzaId": "9A73322488DCB7D9689A6112F2528C9D",
      "participant": "79060002233@cs.us",
      "typeMessage": "locationMessage",
      "location": {
        "nameLocation": "",
        "address": "",
        "jpegThumbnail": "",
        "latitude": 72.5922702,
        "longitude": 45.6645388
      }
    }
  }
}

```

Пример тела уведомления входящего сообщения с изображением и цитатой контакт

```

{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@cs.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1588091580,
  "idMessage": "F7AEC1B7086ECCDC7E6E45923F5EDB825",

```

```

"senderData": {
  "chatId": "79001234568@c.us",
  "sender": "79001234568@c.us",
  "chatName": "Иван",
  "senderName": "Иван",
  "senderContactName": "Иван Васильевич"
},
"messageData": {
  "typeMessage": "imageMessage",
  "fileMessageData": {
    "downloadUrl": "https://sw-media.storage.greenapi.net/542ad819-166b-40a4-b0e1-279069cd03bb.jpeg",
    "caption": "",
    "fileName": "542ad819-166b-b0e1-279069cd03bb.jpeg",
    "jpegThumbnail": "",
    "mimeType": "image/jpeg",
    "forwardingScore": 4,
    "isForwarded": true
  },
  "quotedMessage": {
    "stanzaId": "9A73322488DCB709689A6112F2528C90",
    "participant": "79061230000@c.us",
    "typeMessage": "contactMessage",
    "contact": {
      "displayName": "Green-Api",
      "vcard": "BEGIN:VCARD\nVERSION:3.0\nn:Green-Api\nitem1.TEL;waid=79001230000\nitem1.X-ABLabel:Мобильный\nEND:VCARD"
    }
  }
}
}
}

```

Входящее сообщение с геолокацией

В данном разделе описывается формат входящего уведомления объекта `messageData` для входящего сообщения с геолокацией. Для получения описания общего формата входящих уведомлений обратитесь к разделу [Входящие сообщения](#).

Для получения входящих уведомлений данного вида требуется выполнение двух условий:

```
typeWebhook = incomingMessageReceived
```

```
messageData.typeMessage = locationMessage
```

"Настройка инстанса"

Чтобы получать входящие уведомления о сообщениях с геолокацией, необходимо включить настройки одним из представленных вариантов:

1. Через личный кабинет включить настройку `Получать уведомления о входящих сообщениях и файлах`
2. Через метод `SetSettings` включить настройку `incomingWebhook`

Уведомление Формат уведомления

Поля объекта `messageData`

Параметр	Тип	Описание
<code>typeMessage</code>	string	Тип принятого сообщения. Для сообщений данного типа поле принимает значение <code>locationMessage</code>
<code>locationMessageData</code>	object	Объект данных о принятой геолокации
<code>quotedMessage</code>	object	Объект данных о цитируемом сообщении. Присутствует только, если само сообщение является цитатой

Поля объекта `locationMessageData`

Параметр	Тип	Описание
<code>nameLocation</code>	string	Название локации
<code>address</code>	string	Адрес локации
<code>jpegThumbnail</code>	string	Превью изображения в <code>base64</code> кодировке
<code>latitude</code>	double	Широта локации
<code>longitude</code>	double	Долгота локации
<code>forwardingScore</code>	integer	Количество пересылок сообщения
<code>isForwarded</code>	boolean	Является ли сообщение пересланным, принимает значения <code>true/false</code>

Поля объекта `quotedMessage`

Параметр	Тип	Описание
<code>stanzaId</code>	string	id цитируемого сообщения
<code>participant</code>	string	id отправителя цитируемого сообщения
<code>typeMessage</code>	string	Тип цитируемого сообщения

Остальные поля заполняются в зависимости от типа цитируемого сообщения и идентичны полям входящих сообщений описанных в разделе [Входящие сообщения](#)

Пример тела уведомления

```
{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@с.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1588091580,
  "idMessage": "F7AEC1B7086ECCDC7E6E45923F5EDB825",
  "senderData": {
    "chatId": "79001234568@с.us",
    "sender": "79001234568@с.us",
    "chatName": "Иван",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    "typeMessage": "LocationMessage",
    "locationMessageData": {
      "nameLocation": "Эмпайр-стейт-билдинг",
      "address": "20 W 34th St., New York, NY 10001, США",
      "jpegThumbnail": "/9j/4AAQSkZJRgABAQAAQABAAQ/zwbDAAYEBQYFBAYGBQYHNbYICChAKGk.JChQ0DwwQFw=",
      "latitude": 12.345678910111213,
      "longitude": 14.151617181920212,
      "forwardingScore": 0,
      "isForwarded": false
    }
  }
}
```

Пример тела уведомления входящего сообщения с геопозицией и цитатой текстового сообщения

```
{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@с.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1588091580,
  "idMessage": "F7AEC1B7086ECCDC7E6E45923F5EDB825",
  "senderData": {
    "chatId": "79001234568@с.us",
    "sender": "79001234568@с.us",
    "chatName": "Иван",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    "typeMessage": "LocationMessage",
    "locationMessageData": {
      "nameLocation": "Эмпайр-стейт-билдинг",
      "address": "20 W 34th St., New York, NY 10001, США",
      "jpegThumbnail": "217",
      "latitude": 74.5922641,
      "longitude": 59.6645355,
      "forwardingScore": 4,
      "isForwarded": true
    },
    "quotedMessage": {
      "stanzaId": "9A73322488DCB7D9689A6112F2528C9D",
      "participant": "79001230022@с.us",
      "typeMessage": "textMessage",
      "textMessage": "Привет"
    }
  }
}
```

Входящее сообщение с контактом

В данном разделе описывается формат входящего уведомления объекта `messageData` для входящего сообщения с контактом. Для получения описания общего формата входящих уведомлений обратитесь к разделу [Входящие сообщения](#).

Для получения входящих уведомлений данного вида требуется выполнение двух условий:

```
typeWebhook = incomingMessageReceived
```

```
messageData.typeMessage = contactMessage
```

"Настройка инстанса"

Чтобы получать входящие уведомления о сообщениях с контактом, необходимо включить настройки одним из представленных вариантов:

1. Через личный кабинет включить настройку `Получать уведомления о входящих сообщениях и файлах`
2. Через метод `SetSettings` включить настройку `incomingWebhook`

Уведомление Формат уведомления

Поля объекта `messageData`

Параметр	Тип	Описание
<code>typeMessage</code>	string	Тип принятого сообщения. Для сообщений данного типа поле принимает значение <code>contactMessage</code>
<code>contactMessageData</code>	object	Объект данных о принятом контакте.
<code>quotedMessage</code>	object	Объект данных о цитируемом сообщении. Присутствует только, если само сообщение является цитатой

Поля объекта `contactMessageData`

Параметр	Тип	Описание
<code>displayName</code>	string	Отображаемое имя контакта
<code>vcard</code>	string	Структура VCard (визитной карточки контакта)
<code>isForwarded</code>	boolean	Является ли сообщение пересланным, принимает значения true/false
<code>forwardingScore</code>	integer	Количество пересылок сообщения

Поля объекта `quotedMessage`

Параметр	Тип	Описание
<code>stanzaId</code>	string	id цитируемого сообщения
<code>participant</code>	string	id отправителя цитируемого сообщения
<code>typeMessage</code>	string	Тип цитируемого сообщения

Остальные поля заполняются в зависимости от типа цитируемого сообщения и идентичны полям входящих сообщений описанных в разделе [Входящие сообщения](#)

Пример тела уведомления

```
{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@с.us",
  }
}
```



```

"typeInstance": "whatsapp"
},
"timestamp": 1588091580,
"idMessage": "F7AEC1B7086ECCDC7E6E45923F5EDB825",
"senderData": {
  "chatId": "79001234568@с.us",
  "sender": "79001234568@с.us",
  "chatName": "Иван",
  "senderName": "Иван",
  "senderContactName": "Иван Васильевич"
},
"messageData": {
  "typeMessage": "contactMessage",
  "contactMessageData": {
    "displayName": "Виктор Андреевич",
    "vcard": "BEGIN:VCARD\nVERSION:3.0\nn:Андреевич;Виктор;;\nFN:Виктор Андреевич\nORG:Image\nitem1.TEL;waid=79001234569;+7 900 123-45-69\nitem1.X-ABLabel:Мобильный\nEND:VCARD",
    "forwardingScore": 4,
    "isForwarded": true
  }
}
}
}

```

Пример тела уведомления входящего сообщения с контактом и цитатой текстового сообщения

```

{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@с.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1588091580,
  "idMessage": "F7AEC1B7086ECCDC7E6E45923F5EDB825",
  "senderData": {
    "chatId": "79001234568@с.us",
    "sender": "79001234568@с.us",
    "chatName": "Иван",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    "typeMessage": "contactMessage",
    "contactMessageData": {
      "displayName": "Антиспам",
      "vcard": "BEGIN:VCARD\nVERSION:3.0\nn;;Антиспам;;\nFN:Антиспам\nitem1.TEL:90359362323\nitem1.X-ABLabel:Мобильный\nEND:VCARD",
      "forwardingScore": 4,
      "isForwarded": true
    },
    "quotedMessage": {
      "stanzaId": "9A73322488DCB7D9689A6112F2528C9D",
      "participant": "79001235696@с.us",
      "typeMessage": "textMessage",
      "textMessage": "Привет"
    }
  }
}
}

```

Пример тела уведомления входящего сообщения с контактом и цитатой аудио/видео/документ

```

{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@с.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1588091580,
  "idMessage": "F7AEC1B7086ECCDC7E6E45923F5EDB825",
  "senderData": {
    "chatId": "79001234568@с.us",
    "sender": "79001234568@с.us",
    "chatName": "Иван",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    "typeMessage": "contactMessage",
    "contactMessageData": {
      "displayName": "Антиспам",
      "vcard": "BEGIN:VCARD\nVERSION:3.0\nFN:2 Лена\nitem1.TEL;waid=79001230000\nitem1.X-ABLabel:Мобильный\nEND:VCARD",
      "forwardingScore": 4,
      "isForwarded": true
    }
  },
  "quotedMessage": {
    "stanzaId": "9A73322488DCB7D9689A6112F2528C9D",
    "participant": "79061230000@с.us",
    "typeMessage": "imageMessage",
    "downloadUrl": "",
    "caption": ""
  }
}
}

```

```

    "jpegThumbnail": ""
  }
}

```

Пример тела уведомления входящего сообщения с контактом и цитатой контакта

```

{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@cs.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1588091580,
  "idMessage": "F7AEC1B7086ECCDC7E6E45923F5EDB825",
  "senderData": {
    "chatId": "79001234568@cs.us",
    "sender": "79001234568@cs.us",
    "chatName": "Иван",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    "typeMessage": "contactMessage",
    "contactMessageData": {
      "displayName": "",
      "vcard": "BEGIN:VCARD\nVERSION:3.0\nn:Фонд;\nitem1.TEL;waid=79001203030:/em1.X-ABLabel:Новый тип\nEND:VCARD",
      "forwardingScore": 4,
      "isForwarded": true
    },
    "quotedMessage": {
      "stanzaId": "9A73322488DCB7D9689A6112F2528C9D",
      "participant": "79061230000@cs.us",
      "typeMessage": "contactMessage",
      "contact": {
        "displayName": "Green-Api",
        "vcard": "BEGIN:VCARD\nVERSION:3.0\nn:Green-Api\nitem1.TEL;waid=79001230000\nitem1.X-ABLabel:Мобильный\nEND:VCARD"
      }
    }
  }
}

```

Пример тела уведомления входящего сообщения с контактом и цитатой геопозиции

```

{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@cs.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1588091580,
  "idMessage": "F7AEC1B7086ECCDC7E6E45923F5EDB825",
  "senderData": {
    "chatId": "79001234568@cs.us",
    "sender": "79001234568@cs.us",
    "chatName": "Иван",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    "typeMessage": "contactMessage",
    "contactMessageData": {
      "displayName": "Фонд",
      "vcard": "BEGIN:VCARD\nVERSION:3.0\nFN:2 Сердца\nitem1.TEL;waid=79200000102\nitem1.X-ABLabel:Новый тип\nEND:VCARD",
      "forwardingScore": 4,
      "isForwarded": true
    },
    "quotedMessage": {
      "stanzaId": "9A73322488DCB7D9689A6112F2528C9D",
      "participant": "79060002233@cs.us",
      "typeMessage": "locationMessage",
      "location": {
        "nameLocation": "",
        "address": "",
        "jpegThumbnail": "",
        "latitude": 72.5922702,
        "longitude": 45.6645388
      }
    }
  }
}

```

Входящее сообщение с массивом контактов

В данном разделе описывается формат входящего уведомления объекта `messageData` для входящего сообщения с массивом контактов. Для получения описания общего формата входящих уведомлений обратитесь к разделу [Входящие сообщения](#).

Для получения входящих уведомлений данного вида требуется выполнение двух условий:

```
typeWebhook = incomingMessageReceived
```

```
messageData.typeMessage = contactsArrayMessage
```

"Настройка инстанса"

Чтобы получать входящие уведомления о сообщениях с массивом контактов, необходимо включить настройки одним из представленных вариантов:

1. Через личный кабинет включить настройку `Получать уведомления о входящих сообщениях и файлах`
2. Через метод `SetSettings` включить настройку `incomingWebhook`

Уведомление Формат уведомления

Поля объекта `messageData`

Параметр	Тип	Описание
<code>typeMessage</code>	string	Тип принятого сообщения. Для сообщений данного типа поле принимает значение <code>contactsArrayMessage</code>
<code>contacts</code>	object	Объект массива данных о принятых контактах.
<code>quotedMessage</code>	object	Объект данных о цитируемом сообщении. Присутствует только, если само сообщение является цитатой
<code>isForwarded</code>	boolean	Является ли сообщение пересланным, принимает значения true/false
<code>forwardingScore</code>	integer	Количество пересылок сообщения

Поля объекта `contacts`

Параметр	Тип	Описание
<code>displayName</code>	string	Отображаемое имя контакта
<code>vcard</code>	string	Структура VCard (визитной карточки контакта)

Поля объекта `quotedMessage`

Параметр	Тип	Описание
<code>stanzaId</code>	string	id цитируемого сообщения
<code>participant</code>	string	id отправителя цитируемого сообщения
<code>typeMessage</code>	string	Тип цитируемого сообщения

Остальные поля заполняются в зависимости от типа цитируемого сообщения и идентичны полям входящих сообщений описанных в разделе [Входящие сообщения](#)

Пример тела уведомления

```
{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@с.us",
  }
}
```

```

"typeInstance": "whatsapp"
},
"timestamp": 1683145380,
"idMessage": "9AF4BAC8AA780DD588B0655C247F822D",
"senderData": {
  "chatId": "79001234568@e.us",
  "chatName": "Иван",
  "sender": "79001234568@e.us",
  "senderName": "Иван",
  "senderContactName": "Иван Васильевич"
},
"messageData": {
  "typeMessage": "contactsArrayMessage",
  "messageData": {
    "contacts": [
      {
        "displayName": "Виктор Андреевич",
        "vcard": "BEGIN:VCARD\nVERSION:3.0\nN:Андреевич;Виктор;;\nFN:Виктор Андреевич\nitem1.TEL;waid=79001234567:+7 900 123-45-67\nitem1.X-ABLabel:Мобильный\nEND:VCARD"
      },
      {
        "displayName": "Олег Едросович",
        "vcard": "BEGIN:VCARD\nVERSION:3.0\nN:Едросович;Олег;;\nFN:Олег Едросович\nitem1.TEL;waid=79001234567:+7 900 123-45-67\nitem1.X-ABLabel:Мобильный\nEND:VCARD"
      }
    ]
  },
  "forwardingScore": 0,
  "isForwarded": false
}
}
}

```

Пример тела уведомления входящего сообщения с массивом контактов и цитатой текстового сообщения

```

{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@e.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1683145380,
  "idMessage": "9AF4BAC8AA780DD588B0655C247F822D",
  "senderData": {
    "chatId": "79001234568@e.us",
    "chatName": "Иван",
    "sender": "79001234568@e.us",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    "typeMessage": "contactsArrayMessage",
    "messageData": {
      "contacts": [
        {
          "displayName": "Виктор Андреевич",
          "vcard": "BEGIN:VCARD\nVERSION:3.0\nN:Андреевич;Виктор;;\nFN:Виктор Андреевич\nitem1.TEL;waid=79001234567:+7 900 123-45-67\nitem1.X-ABLabel:Мобильный\nEND:VCARD"
        },
        {
          "displayName": "Олег Едросович",
          "vcard": "BEGIN:VCARD\nVERSION:3.0\nN:Едросович;Олег;;\nFN:Олег Едросович\nitem1.TEL;waid=79001234567:+7 900 123-45-67\nitem1.X-ABLabel:Мобильный\nEND:VCARD"
        }
      ]
    },
    "forwardingScore": 0,
    "isForwarded": false
  },
  "quotedMessage": {
    "stanzaId": "3AA3BE1C2DFB5B8CC315",
    "participant": "79001234567@e.us",
    "typeMessage": "textMessage",
    "textMessage": "Текст"
  }
}
}
}

```

Пример тела уведомления входящего сообщения с массивом контактов и цитатой аудио/видео/документ

```

{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@e.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1683145380,
  "idMessage": "9AF4BAC8AA780DD588B0655C247F822D",
  "senderData": {
    "chatId": "79001234568@e.us",
    "chatName": "Иван",
    "sender": "79001234568@e.us",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },

```

```

"messageData": {
  "typeMessage": "contactsArrayMessage",
  "messageData": {
    "contacts": [
      {
        "displayName": "Виктор Андреевич",
        "vcard": "BEGIN:VCARD\nVERSION:3.0\nn:Андреевич;Виктор;;\nFN:Виктор Андреевич\nitem1.TEL;waid=79001234567:+7 900 123-45-67\nitem1.X-ABLabel:Мобильный\nEND:VCARD"
      },
      {
        "displayName": "Олег Едросович",
        "vcard": "BEGIN:VCARD\nVERSION:3.0\nn:Едросович;Олег;;\nFN:Олег Едросович\nitem1.TEL;waid=79001234567:+7 900 123-45-67\nitem1.X-ABLabel:Мобильный\nEND:VCARD"
      }
    ],
    "forwardingScore": 0,
    "isForwarded": false
  },
  "quotedMessage": {
    "stanzaId": "5E956DF0AC8ED6873FEF",
    "participant": "79001234567@с.us",
    "typeMessage": "imageMessage",
    "downloadUrl": "",
    "caption": "",
    "fileName": "",
    "jpegThumbnail": "/9j/4AAQSkZJRgA"
  }
}
}

```

Пример тела уведомления входящего сообщения с массивом контактов и цитатой контакта

```

{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@с.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1683145380,
  "idMessage": "9AF4BAC8AA780DD588B0655C247F822D",
  "senderData": {
    "chatId": "79001234568@с.us",
    "chatName": "Иван",
    "sender": "79001234568@с.us",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    "typeMessage": "contactsArrayMessage",
    "messageData": {
      "contacts": [
        {
          "displayName": "Виктор Андреевич",
          "vcard": "BEGIN:VCARD\nVERSION:3.0\nn:Андреевич;Виктор;;\nFN:Виктор Андреевич\nitem1.TEL;waid=79001234567:+7 900 123-45-67\nitem1.X-ABLabel:Мобильный\nEND:VCARD"
        },
        {
          "displayName": "Олег Едросович",
          "vcard": "BEGIN:VCARD\nVERSION:3.0\nn:Едросович;Олег;;\nFN:Олег Едросович\nitem1.TEL;waid=79001234567:+7 900 123-45-67\nitem1.X-ABLabel:Мобильный\nEND:VCARD"
        }
      ],
      "forwardingScore": 0,
      "isForwarded": false
    },
    "quotedMessage": {
      "stanzaId": "3AC28E97DA5CA6A31723",
      "participant": "79001234567@с.us",
      "typeMessage": "contactMessage",
      "contact": {
        "displayName": "Green-API",
        "vcard": "BEGIN:VCARD\nVERSION:3.0\nn:И;;\nFN:Green-API\nX-WA-BIZ-NAME:\nX-WA-BIZ-DESCRIPTION:\nTEL;type=CELL;type=VOICE;waid=795800000000:+7 958 000-00-00\nEND:VCARD"
      }
    }
  }
}

```

Пример тела уведомления входящего сообщения с массивом контактов и цитатой геопозиции

```

{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@с.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1683145380,
  "idMessage": "9AF4BAC8AA780DD588B0655C247F822D",
  "senderData": {
    "chatId": "79001234568@с.us",
    "chatName": "Иван",
    "sender": "79001234568@с.us",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  }
}

```

```

},
"messageData": {
  "typeMessage": "contactsArrayMessage",
  "messageData": {
    "contacts": [
      {
        "displayName": "Виктор Андреевич",
        "vcard": "BEGIN:VCARD\nVERSION:3.0\nN:Андреевич;Виктор;;;\nFN:Виктор Андреевич\nitem1.TEL;waid=79001234567:+7 900 123-45-67\nitem1.X-ABLabel:Мобильный\nEND:VCARD"
      },
      {
        "displayName": "Олег Едросович",
        "vcard": "BEGIN:VCARD\nVERSION:3.0\nN:Едросович;Олег;;;\nFN:Олег Едросович\nitem1.TEL;waid=79001234567:+7 900 123-45-67\nitem1.X-ABLabel:Мобильный\nEND:VCARD"
      }
    ],
    "forwardingScore": 0,
    "isForwarded": false
  },
  "quotedMessage": {
    "stanzaId": "3A5473EE888EE52BC20",
    "participant": "79001234567@с.us",
    "typeMessage": "locationMessage",
    "location": {
      "nameLocation": "place",
      "address": "",
      "jpegThumbnail": "",
      "latitude": 52.58900290649414,
      "longitude": 39.659369010253906
    }
  }
}
}
}

```

Входящее сообщение с кнопками

В данном разделе описывается формат входящего уведомления объекта `messageData` для входящего сообщения с кнопками. Для получения описания общего формата входящих уведомлений обратитесь к разделу [Входящие сообщения](#).

Для получения входящих уведомлений данного вида требуется выполнение двух условий:

```
typeWebhook = incomingMessageReceived
```

```
messageData.typeMessage = buttonsMessage
```

"Настройка инстанса"

Чтобы получать входящие уведомления о сообщениях с кнопками, необходимо включить настройки одним из представленных вариантов:

1. Через личный кабинет включить настройку 'Получать уведомления о входящих сообщениях и файлах'
2. Через метод `SetSettings` включить настройку `'incomingWebhook'`

Уведомление Формат уведомления

Поля объекта `messageData`

Параметр	Тип	Описание
<code>typeMessage</code>	string	Тип принятого сообщения. Для сообщений данного типа поле принимает значение <code>buttonsMessage</code>
<code>buttonsMessage</code>	object	Объект данных с кнопками
<code>quotedMessage</code>	object	Объект данных о цитируемом сообщении. Присутствует только, если само сообщение является цитатой

Поля объекта `buttonsMessage`

Параметр	Тип	Описание
<code>contentText</code>	string	Текстовое сообщение тела кнопок
<code>footer</code>	string	Текстовое сообщение подвала кнопок
<code>buttons</code>	object	Объект данных с кнопками
<code>isForwarded</code>	boolean	Является ли сообщение пересланным, принимает значения true/false
<code>forwardingScore</code>	integer	Количество пересылок сообщения

Поля объекта `buttons`

Параметр	Тип	Описание
<code>buttonId</code>	string	id кнопки
<code>buttonText</code>	string	Текст кнопки

Поля объекта `quotedMessage`

Параметр	Тип	Описание
<code>stanzaId</code>	string	id цитируемого сообщения
<code>participant</code>	string	id отправителя цитируемого сообщения
<code>typeMessage</code>	string	Тип цитируемого сообщения

Остальные поля заполняются в зависимости от типа цитируемого сообщения и идентичны полям входящих сообщений описанных в разделе [Входящие сообщения](#)

Пример тела уведомления

```
{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@с.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1588091580,
  "idMessage": "F7AEC1B7086ECDC7E6E45923F5EDB825",
  "senderData": {
    "chatId": "79001234568@с.us",
    "sender": "79001234568@с.us",
    "chatName": "Иван",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    "typeMessage": "buttonsMessage",
    "buttonsMessage": {
      "contentText": "Hello",
      "footer": "Hello",
      "buttons": [
        {
          "buttonId": "1",
          "buttonText": "green"
        },
        {
          "buttonId": "2",
          "buttonText": "red"
        },
        {
          "buttonId": "3",
          "buttonText": "blue"
        }
      ]
    },
    "forwardingScore": 4,
    "isForwarded": true
  }
}
```


Входящее сообщение со списком выбора

В данном разделе описывается формат входящего уведомления объекта `messageData` для входящего сообщения со списком выбора. Для получения описания общего формата входящих уведомлений обратитесь к разделу [Входящие сообщения](#).

Для получения входящих уведомлений данного вида требуется выполнение двух условий:

```
typeWebhook = incomingMessageReceived
```

```
messageData.typeMessage = ListMessage
```

"Настройка инстанса"

Чтобы получать входящие уведомления о сообщениях со списком выбора, необходимо включить настройки одним из представленных вариантов:

1. Через личный кабинет включить настройку 'Получать уведомления о входящих сообщениях и файлах'
2. Через метод `SetSettings` включить настройку `'incomingWebhook'`

Уведомление Формат уведомления

Поля объекта `messageData`

Параметр	Тип	Описание
<code>typeMessage</code>	string	Тип принятого сообщения. Для сообщений данного типа поле принимает значение <code>ListMessage</code>
<code>ListMessage</code>	object	Объект данных со списком
<code>quotedMessage</code>	object	Объект данных о цитируемом сообщении. Присутствует только, если само сообщение является цитатой

Поля объекта `ListMessage`

Параметр	Тип	Описание
<code>contentText</code>	string	Текстовое сообщение тела кнопок
<code>title</code>	string	Нет
<code>footer</code>	string	Текстовое сообщение подвала кнопок
<code>buttonText</code>	string	Нет
<code>sections</code>	array	Да
<code>isForwarded</code>	boolean	Является ли сообщение пересланным, принимает значения true/false
<code>forwardingScore</code>	integer	Количество пересылок сообщения

Поля массива `sections`

Параметр	Тип	Описание
<code>title</code>	string	заголовок списка выбора
<code>rows</code>	array	значения списка выбора

Поля массива `rows`

Параметр	Тип	Описание
<code>title</code>	string	текст значения списка
<code>rowId</code>	string	идентификатор значения списка
<code>description</code>	string	описание значения списка

Поля объекта `quotedMessage`

Параметр	Тип	Описание
<code>stanzaId</code>	string	id цитируемого сообщения
<code>participant</code>	string	id отправителя цитируемого сообщения
<code>typeMessage</code>	string	Тип цитируемого сообщения

Остальные поля заполняются в зависимости от типа цитируемого сообщения и идентичны полям входящих сообщений описанных в разделе [Входящие сообщения](#)

Пример тела уведомления

```
{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@c.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1588091580,
  "idMessage": "F7AEC187086E CDC7E6E45923F5EDB825",
  "senderData": {
    "chatId": "79001234568@c.us",
    "sender": "79001234568@c.us",
    "chatName": "Иван",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    "typeMessage": "ListMessage",
    "ListMessage": {
      "contentText": "Hello",
      "title": "заголовок",
      "footer": "Hello",
      "buttonText": "Список действий",
      "sections": [
        {
          "title": "Секция 1",
          "rows": [
            {
              "title": "Вариант 1",
              "rowId": "option1"
            },
            {
              "title": "Вариант 2",
              "rowId": "option2",
              "description": "Пояснение"
            }
          ]
        },
        {
          "title": "Секция 2",
          "rows": [
            {
              "title": "Опция 3",
              "rowId": "option3"
            },
            {
              "title": "Опция 4",
              "rowId": "option4",
              "description": "Пояснение"
            }
          ]
        }
      ]
    },
    "forwardingScore": 4,
    "isForwarded": true
  }
}
```

Входящее сообщение с шаблонными кнопками

В данном разделе описывается формат входящего уведомления объекта `messageData` для входящего шаблонного сообщения с кнопками. Для получения описания общего формата входящих уведомлений обратитесь к разделу [Входящие сообщения](#).

Для получения входящих уведомлений данного вида требуется выполнение двух условий:

```
typeWebhook = incomingMessageReceived
```

```
messageData.typeMessage = templateMessage
```

"Настройка инстанса"

Чтобы получать входящие уведомления о сообщениях с массивом контактов, необходимо включить настройки одним из представленных вариантов:

1. Через личный кабинет включить настройку 'Получать уведомления о входящих сообщениях и файлах'
2. Через метод `SetSettings` включить настройку `'incomingWebhook'`

Уведомление Формат уведомления

Поля объекта `messageData`

Параметр	Тип	Описание
<code>typeMessage</code>	string	Тип принятого сообщения. Для сообщений данного типа поле принимает значение <code>templateMessage</code>
<code>templateMessage</code>	object	Объект данных с кнопками
<code>quotedMessage</code>	object	Объект данных о цитируемом сообщении. Присутствует только, если само сообщение является цитатой

Поля объекта `templateMessage`

Параметр	Тип	Описание
<code>namespace</code>	string	Пространство имен организации
<code>elementName</code>	string	Имя шаблонного сообщения
<code>contentText</code>	string	Текстовое сообщение тела кнопок
<code>footer</code>	string	Текстовое сообщение подвала кнопок
<code>buttons</code>	object	Объект данных с кнопками
<code>isForwarded</code>	boolean	Является ли сообщение пересланным, принимает значения true/false
<code>forwardingScore</code>	integer	Количество пересылок сообщения

Поля объекта `buttons`

Параметр	Тип	Описание
<code>index</code>	string	Порядок кнопки
<code>urlButton</code>	object	Объект данных url кнопки
<code>callButton</code>	object	Объект данных кнопки обратного звонка
<code>quickReplyButton</code>	object	Объект данных кнопки быстрого ответа

Поля объекта `urlButton`

Параметр	Тип	Описание
<code>displayText</code>	string	Текст url кнопки
<code>url</code>	string	Ссылка

Поля объекта `callButton`

Параметр	Тип	Описание
<code>displayText</code>	string	Текст кнопки обратного звонка
<code>phoneNumber</code>	string	Номер телефона

Поля объекта `quickReplyButton`

Параметр	Тип	Описание
<code>displayText</code>	string	Текст кнопки быстрого ответа
<code>id</code>	string	id кнопки быстрого ответа

Поля объекта `quotedMessage`

Параметр	Тип	Описание
<code>stanzaId</code>	string	id цитируемого сообщения
<code>participant</code>	string	id отправителя цитируемого сообщения
<code>typeMessage</code>	string	Тип цитируемого сообщения

Остальные поля заполняются в зависимости от типа цитируемого сообщения и идентичны полям входящих сообщений описанных в разделе [Входящие сообщения](#)

Пример тела уведомления

```
{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@с.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1588091580,
  "idMessage": "F7AEC1B7086ECCDC7E6E45923F5EDB825",
  "senderData": {
    "chatId": "79001234568@с.us",
    "sender": "79001234568@с.us",
    "chatName": "Иван",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    "typeMessage": "templateMessage",
    "templateMessage": {
      "contentText": "Hello",
      "footer": "Hello",
      "buttons": [
        {
          "urlButton": {
            "displayText": "★ Star us on GitHub!",
            "url": "https://github.com/green-api"
          },
          "index": 1
        },
        {
          "callButton": {
            "displayText": "Call us",
            "phoneNumber": "+1 (234) 5678-901"
          },
          "index": 2
        }
      ]
    }
  }
}
```

```
    },  
    {  
      "quickReplyButton": {  
        "displayText": "Hello",  
        "id": "plainButtonId"  
      },  
      "index": 3  
    }  
  ],  
  "forwardingScore": 4,  
  "isForwarded": true  
} }  
}
```

Входящее сообщение со стикером

В данном разделе описывается формат входящего уведомления объекта `messageData` для входящего сообщения со стикером. Для получения описания общего формата входящих уведомлений обратитесь к разделу [Входящие сообщения](#).

Для получения входящих уведомлений данного вида требуется выполнение двух условий:

```
typeWebhook = incomingMessageReceived
```

```
messageData.typeMessage = stickerMessage
```

"Настройка инстанса"

Чтобы получать входящие уведомления о сообщениях со стикером, необходимо включить настройки одним из представленных вариантов:

1. Через личный кабинет включить настройку `Получать уведомления о входящих сообщениях и файлах`
2. Через метод `SetSettings` включить настройку `incomingWebhook`

Уведомление Формат уведомления

Поля объекта `messageData`

Параметр	Тип	Описание
<code>typeMessage</code>	string	Тип принятого сообщения. Для сообщений данного типа поле принимает значение: <code>stickerMessage</code>
<code>fileMessageData</code>	object	Объект данных о принятом стикере
<code>quotedMessage</code>	object	Объект данных о цитируемом сообщении

Поля объекта `fileMessageData`

Параметр	Тип	Описание
<code>downloadUrl</code>	string	Ссылка на стикер
<code>caption</code>	string	Описание под картинкой или видео
<code>fileName</code>	string	Название файла
<code>jpegThumbnail</code>	string	Предпросмотр изображения в base64
<code>isAnimated</code>	boolean	Является ли стикер анимированным
<code>mimeType</code>	string	Тип файла, согласно классификации Media Types
<code>isForwarded</code>	boolean	Является ли сообщение пересланным, принимает значения true/false
<code>forwardingScore</code>	integer	Количество пересылок сообщения

Поля объекта `quotedMessage`

Параметр	Тип	Описание
<code>stanzaId</code>	string	id цитируемого сообщения
<code>participant</code>	string	id отправителя цитируемого сообщения
<code>typeMessage</code>	string	Тип цитируемого сообщения

Остальные поля заполняются в зависимости от типа цитируемого сообщения и идентичны полям входящих сообщений описанных в разделе [Входящие сообщения](#)

Пример тела уведомления

```
{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@c.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1588091580,
  "idMessage": "F7AEC187086E CDC7E6E45923F5EDB825",
  "senderData": {
    "chatId": "79001234568@c.us",
    "chatName": "Иван",
    "sender": "79001234568@c.us",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    "typeMessage": "stickerMessage",
    "fileMessageData": {
      "downloadUrl": "https://sw-media.storage.greenapi.net/9901742665/607dde0a-01dc-4dd9-8afb-707cbf610943.webp",
      "caption": "",
      "fileName": "201ee6e7-b6f9-48fe-bc5a-1ce4341cce88.webp",
      "jpegThumbnail": "",
      "isAnimated": true,
      "mimeType": "image/webp",
      "forwardingScore": 1,
      "isForwarded": true
    },
    "quotedMessage": {
      "stanzaId": "A16014FAC9F5382E7B9CEEF1FFEB6F",
      "participant": "79001234568@c.us"
    }
  }
}
```

Входящее сообщение-реакция

В данном разделе описывается формат входящего уведомления объекта `messageData` для входящего сообщения реакции. Для получения описания общего формата входящих уведомлений обратитесь к разделу [Входящие сообщения](#).

Для получения входящих уведомлений данного вида требуется выполнение двух условий:

```
typeWebhook = incomingMessageReceived
```

```
messageData.typeMessage = reactionMessage
```

"Настройка инстанса"

Чтобы получать входящие уведомления о сообщениях-реакциях, необходимо включить настройки одним из представленных вариантов:

1. Через личный кабинет включить настройку `Получать уведомления о входящих сообщениях и файлах`
2. Через метод `SetSettings` включить настройку `incomingWebhook`

Уведомление Формат уведомления

Поля объекта `messageData`

Параметр	Тип	Описание
<code>typeMessage</code>	string	Тип принятого сообщения. Для сообщений данного типа поле принимает значение: <code>reactionMessage</code>
<code>extendedTextMessageData</code>	object	Объект данных о принятом сообщении-реакции
<code>quotedMessage</code>	object	Объект данных о цитируемом сообщении. Сообщение на которое отреагировал собеседник

Поля объекта `extendedTextMessageData`

Параметр	Тип	Описание
<code>text</code>	string	Реакция (эмоджи) на сообщение

Поля объекта `quotedMessage`

Параметр	Тип	Описание
<code>stanzaId</code>	string	id цитируемого сообщения
<code>participant</code>	string	id отправителя цитируемого сообщения

Остальные поля заполняются в зависимости от типа цитируемого сообщения и идентичны полям входящих сообщений описанных в разделе [Входящие сообщения](#)

Пример тела уведомления

```
{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@с.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1588091580,
  "idMessage": "F7AEC1B7086ECCDC7E6E45923F5EDB825",
  "senderData": {
    "chatId": "79001234568@с.us",
    "chatName": "Иван",
    "sender": "79001234568@с.us",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  }
}
```



```
"messageData": {  
  "typeMessage": "reactionMessage",  
  "extendedTextMessageData": {  
    "text": "👍"  
  },  
  "quotedMessage": {  
    "stanzaId": "A16014FAC9F5382E789CEEEFF1FFEB6F",  
    "participant": "79001234568@c.us"  
  }  
}
```

Входящее сообщение приглашение в группу

В данном разделе описывается формат входящего уведомления объекта `messageData` для входящего сообщения приглашения в группу. Для получения описания общего формата входящих уведомлений обратитесь к разделу [Входящие сообщения](#).

Для получения входящих уведомлений данного вида требуется выполнение двух условий:

```
typeWebhook = incomingMessageReceived
```

```
messageData.typeMessage = groupInviteMessage
```

"Условия получения уведомления типа `groupInviteMessage`"

Чтобы получать входящие уведомления о приглашении в группу, необходимо включить настройки одним из представленных вариантов:

1. Через личный кабинет включить настройки `Получать уведомления о входящих сообщениях и файлах`
2. Через метод `SetSettings` включить настройки `incomingWebhook`

Условия получения уведомления типа `groupInviteMessage`:

- номер телефона отправителя ****сохранён**** у получателя в контактах;
- номер телефона получателя ****не сохранён**** у отправителя в контактах;
- у получателя в настройках конфиденциальности возможность отправлять приглашения в группу настроена только для контактов.

Для отправки приглашения в `новую группу` необходимо:

1. Создать группу WhatsApp с номером получателя
2. Вы получаете уведомление `"Не удалось добавить номер <...>".` Вы можете пригласить контакт в группу лично"
3. Выбрать `"Пригласить в группу"` в данном уведомлении.

Для отправки приглашения в `существующую группу`:

1. Зайти в существующую группу, открыть "данные группы", выбрать "добавить участников", выбрать получателя уведомления
2. Вы получаете уведомление `"Не удалось добавить номер <...>".` Вы можете пригласить контакт в группу лично"
3. Выбрать `"Пригласить в группу"` в данном уведомлении.

Таким образом получателю придёт приглашение в группу и уведомление `groupInviteMessage`` на связанный инстанс.
> Ссылки на приглашение в группу не являются сообщениями с типом `groupInviteMessage``.

> Получение данного типа уведомления возможно, только при отправке приглашения через устройство. Отправка данного приглашения через API невозможна.

Уведомление Формат уведомления

Поля объекта `messageData`

Параметр	Тип	Описание
<code>typeMessage</code>	string	Тип принятого сообщения. Для сообщений данного типа поле принимает значение: <code>groupInviteMessage</code>
<code>groupInviteMessageData</code>	object	Объект данных о принятом сообщении приглашении в группу
<code>quotedMessage</code>	object	Объект данных о цитируемом сообщении.

Поля объекта `groupInviteMessageData`

Параметр	Тип	Описание
<code>groupJid</code>	string	<code>chatId</code> Группы
<code>inviteCode</code>	string	Код приглашения
<code>inviteExpiration</code>	string	Срок действия приглашения
<code>groupName</code>	string	Название группы
<code>caption</code>	string	Описание сообщения
<code>name</code>	string	Имя отправителя
<code>jpegThumbnail</code>	string	Предпросмотр изображения в base64

Поля объекта `quotedMessage`

Параметр	Тип	Описание
<code>stanzaId</code>	string	id цитируемого сообщения
<code>participant</code>	string	id отправителя цитируемого сообщения
<code>typeMessage</code>	string	Тип цитируемого сообщения

Остальные поля заполняются в зависимости от типа цитируемого сообщения и идентичны полям входящих сообщений описанных в разделе [Входящие сообщения](#)

Пример тела уведомления

```
{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@c.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1588091580,
  "idMessage": "F7AEC1B7086ECCDC7E6E45923F5EDB825",
  "senderData": {
    "chatId": "79001234568@c.us",
    "chatName": "Иван",
    "sender": "79001234568@c.us",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    "typeMessage": "groupInviteMessage",
    "groupInviteMessageData": {
      "groupJid": "79099197688-1506012221@g.us",
      "inviteCode": "a7E5WU/g7rmjaQnv",
      "inviteExpiration": "0",
      "groupName": "Махловка, рисование 4-6",
      "caption": "Приглашение в мою группу WhatsApp",
      "name": "myname",
      "jpegThumbnail": "/9j/4AAQSkZJRgABAQAAQABAAQ/zwBDAAYEBQY/9k="
    }
  }
}
```

Входящее сообщение опроса

В данном разделе описывается формат входящего уведомления объекта `messageData` для входящего сообщения опроса. Для получения описания общего формата входящих уведомлений обратитесь к разделу [Входящие сообщения](#).

Для получения входящих уведомлений данного вида требуется выполнение двух условий:

```
typeWebhook = incomingMessageReceived
```

```
messageData.typeMessage = pollMessage
```

"Настройка инстанса"

Чтобы получать входящие уведомления о сообщениях с опросом, необходимо включить настройки одним из представленных вариантов:

1. Через личный кабинет включить настройки 'Получать уведомления о входящих сообщениях и файлах' и 'Получать уведомления об опросах'
2. Через метод `SetSettings` включить настройки `'incomingWebhook'` и `'pollMessageWebhook'`

Уведомление Формат уведомления

Поля объекта `messageData`

Параметр	Тип	Описание
<code>typeMessage</code>	string	Тип принятого сообщения. Для сообщений данного типа поле принимает значение <code>pollMessage</code>
<code>pollMessageData</code>	object	Объект данных о сообщении опроса

Поля объекта `pollMessageData`

Параметр	Тип	Описание
<code>name</code>	string	Название опроса
<code>options</code>	array	Массив данных о вариантах выбора опроса
<code>multipleAnswers</code>	boolean	Разрешить несколько ответов. <code>true</code> — включено, <code>false</code> — отключено, По умолчанию: <code>false</code>

Поля массива `options`

Параметр	Тип	Описание
<code>optionName</code>	string	Название варианта выбора

Пример тела уведомления

```
{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@с.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1588091580,
  "idMessage": "F7AEC1B7086ECCDC7E6E45923F5EDB825",
  "senderData": {
    "chatId": "79001234568@с.us",
    "sender": "79001234568@с.us",
    "chatName": "Иван",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    "typeMessage": "pollMessage",
    "pollMessageData": {
      "name": "Poll Name",
      "options": [
        {

```

```
    "optionName": "Variant 1"  
  },  
  {  
    "optionName": "Variant 2"  
  }  
],  
"multipleAnswers": false  
}  
}
```

Входящее сообщение с обновлением опроса

В данном разделе описывается формат входящего уведомления объекта `messageData` для обновления входящего сообщения опроса. Для получения описания общего формата входящих уведомлений обратитесь к разделу [Входящие сообщения](#).

Для получения входящих уведомлений данного вида требуется выполнение двух условий:

```
typeWebhook = incomingMessageReceived
```

```
messageData.typeMessage = pollUpdateMessage
```

"Настройка инстанса"

Чтобы получать входящие уведомления об обновлениях опроса, необходимо включить настройки одним из представленных вариантов:

1. Через личный кабинет включить настройки 'Получать уведомления о входящих сообщениях и файлах' и 'Получать уведомления об опросах'
2. Через метод `SetSettings` включить настройки `'incomingWebhook'` и `'pollMessageWebhook'`
 - >- Если выбрать вариант в опросе, затем сразу его отменить, то вебхук `'pollUpdateMessage'` придет по последнему состоянию опроса.
 - >- Если выбрать вариант в опросе, затем его отменить спустя секунду или позже, то вебхук `'pollUpdateMessage'` придет по обоим состояниям опроса.

Уведомление Формат уведомления

Поля объекта `messageData`

Параметр	Тип	Описание
<code>typeMessage</code>	string	Тип принятого сообщения. Для сообщений данного типа поле принимает значение <code>pollUpdateMessage</code>
<code>pollMessageData</code>	object	Объект данных о сообщении опроса

Поля объекта `pollMessageData`

Параметр	Тип	Описание
<code>stanzaId</code>	string	id сообщения с опросом
<code>name</code>	string	Название опроса
<code>votes</code>	array	Массив данных голосования по опросу
<code>multipleAnswers</code>	boolean	Разрешить несколько ответов. <code>true</code> — включено, <code>false</code> — отключено, По умолчанию: <code>false</code>

Поля массива `votes`

Параметр	Тип	Описание
<code>optionName</code>	string	Название варианта выбора
<code>optionVoters</code>	array	Массив wid пользователей выбравшие этот вариант

Поля массива `optionVoters`

Параметр	Тип	Описание
<code>default</code>	string	Wid пользователя, выбравшего опцию

Пример тела уведомления

```
{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
```

```
"wid": "11001234567@c.us",
"typeInstance": "whatsapp"
},
"timestamp": 1588091580,
"idMessage": "F7AEC1B7086E CDC7E6E45923F5EDB825",
"senderData": {
  "chatId": "79001234568@c.us",
  "sender": "79001234568@c.us",
  "chatName": "Иван",
  "senderName": "Иван",
  "senderContactName": "Иван Васильевич"
},
"messageData": {
  "typeMessage": "pollUpdateMessage",
  "pollMessageData": {
    "stanzaId": "BAE53AFDD5F0C000",
    "name": "Poll Name",
    "votes": [
      {
        "optionName": "Variant 1",
        "optionVoters": []
      },
      {
        "optionName": "Variant 2",
        "optionVoters": ["79001234568@c.us"]
      }
    ]
  },
  "multipleAnswers": false
}
}
```

(не используется) Сообщение с цитатой

Данное сообщение устарело и оставлено для обратной совместимости. Для замены смотрите поле `quotedMessage` в других типах входящих сообщений

В данном разделе описывается формат входящего уведомления объекта `messageData` для входящего сообщения с цитатой. Для получения описания общего формата входящих уведомлений обратитесь к разделу [Входящие сообщения](#).

Для получения входящих уведомлений данного вида требуется выполнение двух условий:

```
typeWebhook = incomingMessageReceived
```

```
messageData.typeMessage = quotedMessage
```

"Настройка инстанса"

Чтобы получать входящие уведомления о сообщениях с массивом контактов, необходимо включить настройки одним из представленных вариантов:

1. Через личный кабинет включить настройку `Получать уведомления о входящих сообщениях и файлах`
2. Через метод `SetSettings` включить настройку `incomingWebhook``

Уведомление Формат уведомления

Поля объекта `messageData`

Параметр	Тип	Описание
<code>typeMessage</code>	string	Тип принятого сообщения. Для сообщений данного типа поле принимает значение <code>quotedMessage</code>
<code>extendedTextMessageData</code>	object	Объект данных о текстовом сообщении

Поля объекта `extendedTextMessageData`

Параметр	Тип	Описание
<code>text</code>	string	Текстовое сообщение под цитируемым
<code>stanzaId</code>	string	ID цитируемого сообщения
<code>participant</code>	string	ID чата получателя

Пример тела уведомления

```
{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "70009876543@cc.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1588091580,
  "idMessage": "F7AEC1B7086ECCDC7E6E45923F5EDB825",
  "senderData": {
    "chatId": "79001234568@cc.us",
    "sender": "79001234568@cc.us",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    "typeMessage": "quotedMessage",
    "extendedTextMessageData": {
      "text": "Цитируем это",
      "stanzaId": "3EB0D70028C312EB513C",
      "participant": "70009876543@cc.us"
    }
  }
}
```


ВЫБОР КНОПОК

Выбор обычной кнопки

В данном разделе описывается формат входящего уведомления объекта `messageData` для обычных кнопок. Для получения описания общего формата входящих уведомлений обратитесь к разделу [Входящие сообщения](#).

Уведомление Формат уведомления

Поля объекта `messageData`

Параметр	Тип	Описание
<code>typeMessage</code>	string	Тип принятого сообщения. Для сообщений данного типа поле принимает значение <code>buttonsResponseMessage</code>
<code>buttonsResponseMessage</code>	object	Объект данных о нажатии пользователя на кнопку

Поля объекта `buttonsResponseMessage`

Параметр	Тип	Описание
<code>selectedButtonId</code>	string	идентификатор нажатой кнопки
<code>selectedButtonText</code>	string	текст нажатой кнопки
<code>stanzaId</code>	string	ID сообщения с кнопками

Пример тела уведомления

```
{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": "1101000001",
    "wid": "11001234567@ec.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": "1656315272",
  "idMessage": "9BB19BB0D568BA7B185EEAD21A33D317",
  "senderData": {
    "chatId": "11001234567@ec.us",
    "sender": "11001234567@ec.us",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    "typeMessage": "buttonsResponseMessage",
    "buttonsResponseMessage": {
      "stanzaId": "BAE53AFDD05F0C137",
      "selectedButtonId": "1",
      "selectedButtonText": "Green"
    }
  }
}
```

Выбор шаблонной кнопки

В данном разделе описывается формат входящего уведомления объекта `messageData` для шаблонных кнопок. Для получения описания общего формата входящих уведомлений обратитесь к разделу [Входящие сообщения](#).

Уведомление Формат уведомления

Поля объекта `messageData`

Параметр	Тип	Описание
<code>typeMessage</code>	string	Тип принятого сообщения. Для сообщений данного типа поле принимает значение <code>templateButtonsReplyMessage</code>
<code>templateButtonReplyMessage</code>	object	Объект данных о нажатии пользователя на кнопку

Поля объекта `templateButtonReplyMessage`

Параметр	Тип	Описание
<code>selectedIndex</code>	string	числовой индекс нажатой кнопки
<code>selectedId</code>	string	идентификатор нажатой кнопки
<code>selectedDisplayText</code>	string	текст нажатой кнопки
<code>stanzaId</code>	string	ID сообщения с кнопками

Пример тела уведомления

```
{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": "1101000001",
    "wid": "11001234567@ec.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": "1656315272",
  "idMessage": "9BB19BB0D568BA7B185EEAD21A33D317",
  "senderData": {
    "chatId": "11001234567@ec.us",
    "sender": "11001234567@ec.us",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    "typeMessage": "templateButtonReplyMessage",
    "templateButtonReplyMessage": {
      "stanzaId": "BAE53AFDD5F0C137",
      "selectedIndex": "2",
      "selectedId": "id1",
      "selectedDisplayText": "red"
    }
  }
}
```

Выбор элемента списка

В данном разделе описывается формат входящего уведомления объекта `messageData` для списка выбора. Для получения описания общего формата входящих уведомлений обратитесь к разделу [Входящие сообщения](#).

Уведомление Формат уведомления

Поля объекта `messageData`

Параметр	Тип	Описание
<code>typeMessage</code>	string	Тип принятого сообщения. Для сообщений данного типа поле принимает значение <code>listResponseMessage</code>
<code>listResponseMessage</code>	object	Объект данных о нажатии пользователя на значение списка
<code>quotedMessage</code>	object	Объект данных о цитируемом сообщении. Присутствует только, если само сообщение является цитатой

Поля объекта `listResponseMessage`

Параметр	Тип	Описание
<code>title</code>	string	текст выбранного значения
<code>listType</code>	string	тип списка. 1 - выбор одного значения
<code>singleSelectReply</code>	string	id выбранного значения
<code>stanzaId</code>	string	ID сообщения с кнопками

Пример тела уведомления

```
{
  "typeWebhook": "incomingMessageReceived",
  "instanceData": {
    "idInstance": "1101000001",
    "wid": "11001234567@с.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": "1657025112",
  "idMessage": "A36C68A09C54796295FCAD20C15C534C",
  "senderData": {
    "chatId": "11001234567@с.us",
    "sender": "11001234567@с.us",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    "typeMessage": "ListResponseMessage",
    "listResponseMessage": {
      "stanzaId": "BAE53AFDD5F0C137",
      "title": "Вариант 2",
      "listType": "1",
      "singleSelectReply": "option2"
    }
  }
}
```

ОТПРАВЛЕННОЕ СООБЩЕНИЕ

Отправленное с телефона сообщение

Формат сообщения, отправленного с телефона, идентичен формату [входящего сообщения](#), при этом [тип входящего уведомления](#) принимает значение `outgoingMessageReceived`.

"Настройка инстанса"

Чтобы получать входящие уведомления о сообщениях, отправленных с телефона, необходимо включить настройки одним из представленных вариантов:

1. Через личный кабинет включить настройку `Получать уведомления о сообщениях, отправленных с телефона` (дополнительно `Получать уведомления о статусах отправки/доставки/прочтении исходящих сообщений` для получения уведомлений о статусах отправленного сообщения)
2. Через метод `SetSettings` включить настройку ``outgoingMessageWebhook`` (дополнительно ``outgoingWebhook`` для получения уведомлений о статусах отправленного сообщения)

Пример тела уведомления

```
{
  "typeWebhook": "outgoingMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@ec.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1588091580,
  "idMessage": "F7AEC1B7086ECDC7E6E45923F5EDB825",
  "senderData": {
    "chatId": "79001234568@ec.us",
    "sender": "79001234568@ec.us",
    "chatName": "Коля",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    // В зависимости от typeMessage = textMessage || imageMessage || videoMessage || documentMessage || audioMessage || locationMessage || contactMessage || extendedTextMessage ||
    pollMessage
    ...
    ...
    ...
  }
}
```

Отправленное сообщение через API

Формат сообщения, отправленного с API, идентичен формату **входящего сообщения**, при этом **тип входящего уведомления** принимает значение `outgoingAPIMessageReceived`.

"Настройка инстанса"

Чтобы получать входящие уведомления об отправленных через API сообщениях, необходимо включить настройки одним из представленных вариантов:

1. Через личный кабинет включить настройку `Получать уведомления о сообщениях, отправленных через API` (дополнительно `Получать уведомления о статусах отправки/доставки/прочтении исходящих сообщений` для получения уведомлений о статусах отправленного сообщения)
2. Через метод `SetSettings` включить настройку `outgoingAPIMessageWebhook` (дополнительно outgoingWebhook` для получения уведомлений о статусах отправленного сообщения)`

Пример тела уведомления

```
{
  "typeWebhook": "outgoingAPIMessageReceived",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@cc.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1588091580,
  "idMessage": "F7AEC1B7086ECCDC7E6E45923F5EDB825",
  "senderData": {
    "chatId": "79001234568@cc.us",
    "sender": "79001234568@cc.us",
    "chatName": "Коля",
    "senderName": "Иван",
    "senderContactName": "Иван Васильевич"
  },
  "messageData": {
    // В зависимости от typeMessage = textMessage || imageMessage || videoMessage || documentMessage || audioMessage || locationMessage || contactMessage || extendedTextMessage ||
    pollMessage
    ...
    ...
    ...
  }
}
```

Статус отправленного сообщения

Входящее уведомление данного типа содержит статус ранее отправленного сообщения: отправлено, доставлено, прочитано и др.

"Настройка инстанса"

Чтобы получать входящие уведомления о статусах отправленных сообщений, необходимо включить настройки одним из представленных вариантов:

1. Через личный кабинет включить настройки `Получать уведомления о сообщениях, отправленных с телефона`, `Получать уведомления о сообщениях, отправленных через API` и `Получать уведомления о статусах отправки/доставки/прочтении исходящих сообщений`
2. Через метод `SetSettings` включить настройки `outgoingMessageWebhook`, `outgoingAPIMessageWebhook` и `outgoingWebhook`

Подробнее о статусах отправленных сообщений в WhatsApp читайте в нашей [статье](#)

Уведомление Формат уведомления

Параметр	Тип	Описание
<code>typeWebhook</code>	string	Тип входящего уведомления. Для уведомлений данного типа поле принимает значение <code>outgoingMessageStatus</code>
<code>chatId</code>	string	Идентификатор чата. Чат с сообщением по которому получили статус.
<code>instanceData</code>	object	Данные об инстансе
<code>timestamp</code>	integer	Время наступления события в UNIX-формате
<code>idMessage</code>	string	Идентификатор отправленного сообщения или файла. Идентификатор отправленного сообщения возвращается методами: <code>SendMessage</code> , <code>SendFileByUrl</code> , <code>SendFileByUpload</code> , <code>SendLocation</code> , <code>SendContact</code> , <code>SendPoll</code>
<code>status</code>	string	Статус отправленного сообщения или файла. Статус принимает значения: <ul style="list-style-type: none"> <code>sent</code> - сообщение отправлено <code>delivered</code> - сообщение доставлено до получателя <code>read</code> - сообщение прочитано/просмотрено/прослушано <code>failed</code> - произошла ошибка при отправке сообщения на сервер WhatsApp (данный статус невозможно отключить в настройках <code>SetSettings</code>, необходимо обязательно реализовать обработку данного уведомления) <code>noAccount</code> - на номере телефона получателя не зарегистрирован аккаунт WhatsApp (данный статус невозможно отключить в настройках <code>SetSettings</code>, необходимо обязательно реализовать обработку данного уведомления) <code>notInGroup</code> - отправитель не является участником группового чата, в который выполняется отправка сообщения <code>yellowCard</code> - приостановка отправки сообщений из-за спамерской активности. Сообщения отправленные после получения статуса хранятся в очереди к отправке 24 часа. Для продолжения работы инстанса требуется сделать перезагрузку инстанса
<code>description</code>	string	Описание ошибки (рекомендуется сделать информирование оператора об ошибках системы)
<code>sendByApi</code>	boolean	Флаг указывает отправлено сообщение из API или нет, принимает значения: <code>true</code> , <code>false</code>

Поля объекта `instanceData`

Параметр	Тип	Описание
<code>idInstance</code>	integer	Идентификатор инстанса. Размер <code>integer</code> равен <code>int64</code> . Значения лежат в диапазоне от 1 до 10 разрядов
<code>wid</code>	string	Идентификатор аккаунта в формате WhatsApp
<code>typeInstance</code>	string	Тип мессенджера для инстанса

Пример тела уведомления со статусом `sent`

```
{
  "typeWebhook": "outgoingMessageStatus",
  "chatId": "11001234567890@c.us",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@c.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1727691478,
  "idMessage": "3EB0608D6A2901063D63",
  "status": "sent",
  "sendByApi": true
}
```

Пример тела уведомления со статусом `delivered`

```
{
  "typeWebhook": "outgoingMessageStatus",
  "chatId": "11001234567890@c.us",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@c.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1727698517,
  "idMessage": "3EB0608D6A2901063D63",
  "status": "delivered",
  "sendByApi": true
}
```

Пример тела уведомления со статусом `read`

```
{
  "typeWebhook": "outgoingMessageStatus",
  "chatId": "11001234567890@c.us",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@c.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1727698597,
  "idMessage": "3EB0608D6A2901063D63",
  "status": "read",
  "sendByApi": true
}
```

Пример тела уведомления при отправке в группу, в которой не состоит отправитель

```
{
  "typeWebhook": "outgoingMessageStatus",
  "chatId": "120363347693198097@g.us",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@c.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1727760222,
  "idMessage": "BAE5245994546A06",
  "status": "failed",
  "description": "Error: item-not-found",
  "sendByApi": true
}
```

Пример тела уведомления при отсутствии аккаунта WhatsApp на номере

```
{
  "typeWebhook": "outgoingMessageStatus",
  "chatId": "11001234567890@c.us",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@c.us",
  }
}
```

```
    "typeInstance": "whatsapp"
  },
  "timestamp": 1727759586,
  "idMessage": "3EB0608D6A2901063D63",
  "status": "noAccount",
  "sendByApi": true
}
```

Пример тела уведомления с ошибкой SWE004

```
{
  "typeWebhook": "outgoingMessageStatus",
  "chatId": "11001234567890@ec.us",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@ec.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1586700802,
  "idMessage": "3EB0608D6A2901063D63",
  "status": "failed",
  "description": "SWE004: the number of participants in the group has been exceeded - 706", // 706 - количество участников в группе
  "sendByApi": true
}
```


ПРОЧИЕ

Статус инстанса

Входящее уведомление данного типа содержит данные о состоянии авторизации инстанса

"Настройка инстанса"

Чтобы получать входящие уведомления о сообщениях с геолокацией, необходимо включить настройки одним из представленных вариантов:

1. Через личный кабинет включить настройку `Получать уведомления об изменении состояния авторизации аккаунта`
2. Через метод `SetSettings` включить настройку `stateWebhook``

Уведомление Формат уведомления

Параметр	Тип	Описание
<code>typeWebhook</code>	string	Тип входящего уведомления. Для уведомлений данного типа поле принимает значение <code>stateInstanceChanged</code>
<code>instanceData</code>	object	Данные об инстансе
<code>timestamp</code>	integer	Время наступления события в UNIX-формате
<code>stateInstance</code>	string	Состояние инстанса. Принимает значения:
		<code>notAuthorized</code> - Инстанс не авторизован. Для авторизации инстанса обратитесь к разделу Перед началом работы
		<code>authorized</code> - Инстанс авторизован
		<code>blocked</code> - Инстанс забанен
		<code>sleepMode</code> - Статус устарел. Инстанс ушел в спящий режим. Состояние возможно, когда выключен телефон. После включения телефона может потребоваться до 5 минут для перевода состояния инстанса в значение <code>authorized</code> .
		<code>starting</code> - Инстанс в процессе запуска (сервисный режим). Происходит перезагрузка инстанса, сервера или инстанс в режиме обслуживания. Может потребоваться до 5 минут для перевода состояния инстанса в значение <code>authorized</code> .
		<code>yellowCard</code> - На инстансе частично или полностью приостановлена отправка сообщений из-за спамерской активности. Сообщения отправленные после получения статуса хранятся в очереди к отправке 24 часа. Для продолжения работы инстанса требуется сделать перезагрузку инстанса

Поля объекта `instanceData`

Параметр	Тип	Описание
<code>idInstance</code>	integer	Идентификатор инстанса. Размер <code>integer</code> равен <code>int64</code> . Значения лежат в диапазоне от 1 до 10 разрядов
<code>wid</code>	string	Идентификатор аккаунта в формате WhatsApp
<code>typeInstance</code>	string	Тип мессенджера для инстанса

Пример тела уведомления

```
{
  "typeWebhook": "stateInstanceChanged",
  "instanceData": {
```

```
"idInstance": 1,  
"wid": "",  
"typeInstance": "whatsapp"  
},  
"timestamp": 1586700690,  
"stateInstance": "notAuthorized"  
}
```

Аватар контакта

"Уведомление временно не работает."

Входящее уведомление данного типа содержит данные об аватаре контакта по запрошенному номеру телефона

Уведомление Формат уведомления

Параметр	Тип	Описание
typeWebhook	string	Тип входящего уведомления. Для уведомлений данного типа поле принимает значение deviceInfo
instanceData	object	Данные об инстансе
timestamp	integer	Время наступления события в UNIX-формате
deviceData	object	Данные об устройстве

Поля объекта instanceData

Параметр	Тип	Описание
idInstance	integer	Идентификатор инстанса. Размер integer равен int64. Значения лежат в диапазоне от 1 до 10 разрядов
wid	string	Идентификатор аккаунта в формате WhatsApp
typeInstance	string	Тип мессенджера для инстанса

Поля объекта avatarInfo

Параметр	Тип	Описание
chatId	string	Идентификатор чата, для которого получен аватар контакта
urlAvatar	string	Ссылка на аватар корреспондента или группового чата. Параметр принимает пустое значение в случае, если аватар не установлен
reason	string	Причина почему аватар не был проверен. Присутствует когда не удалось выполнить проверку, возможные значения: bad request data - Неверный формат номера телефона. Номер телефона должен содержать 11 или 12 цифр. Или идентификатора чата get avatar timeout limit exceeded - Превышен лимит времени ожидания ответа о наличии аватара

Пример тела уведомления

```
{
  "typeWebhook": "deviceInfo",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@c.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1586700802,
  "avatarInfo": {
    "chatId": "79001234560@c.us",
    "urlAvatar": "https://pps.whatsapp.net/v/t61.24694-24/180589560_n.jpg?ccb=11-4&oh=f3fb1&oe=615A60EB"
  }
}
```

Статус сокета инстанса

Входящее уведомление данного типа содержит данные о состоянии сокета (готов или нет инстанс отправлять/принимать сообщения)

"Настройка инстанса"

Чтобы получать входящие уведомления о сообщениях с геолокацией, необходимо включить настройки одним из представленных вариантов:

1. Через личный кабинет включить настройку `Получать уведомления об изменении состояния авторизации аккаунта`
2. Через метод `SetSettings` включить настройку `stateWebhook``

Уведомление Формат уведомления

Параметр	Тип	Описание
<code>typeWebhook</code>	string	Тип входящего уведомления. Для уведомлений данного типа поле принимает значение <code>statusInstanceChanged</code>
<code>instanceData</code>	object	Данные об инстансе
<code>timestamp</code>	integer	Время наступления события в UNIX-формате
<code>statusInstance</code>	string	Состояние инстанса. Принимает значения:
		<code>online</code> - Инстанс может принимать/отправлять сообщения, сокет открыт
		<code>offline</code> - Инстанс не может принимать/отправлять сообщения, сокет закрыт

Поля объекта `instanceData`

Параметр	Тип	Описание
<code>idInstance</code>	integer	Идентификатор инстанса. Размер <code>integer</code> равен <code>int64</code> . Значения лежат в диапазоне от 1 до 10 разрядов
<code>wid</code>	string	Идентификатор аккаунта в формате WhatsApp
<code>typeInstance</code>	string	Тип мессенджера для инстанса

Пример тела уведомления

```
{
  "typeWebhook": "statusInstanceChanged",
  "instanceData": {
    "idInstance": 1,
    "wid": "11001234567@ec.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1586700690,
  "statusInstance": "onLine"
}
```

Входящий звонок

Уведомление данного типа возникает при входящем звонке и содержит информацию об инициаторе и адресате звонка.

"Настройка инстанса"

Чтобы получать входящие уведомления о входящих звонках, необходимо включить настройки одним из представленных вариантов:

1. Через личный кабинет включить настройку `Получать уведомления о звонках`
2. Через метод `SetSettings` включить настройку ``incomingCallWebhook``

Уведомление Формат уведомления

Параметр	Тип	Описание
<code>from</code>	string	Идентификатор инициатора звонка
<code>typeWebhook</code>	string	Тип входящего уведомления. Для уведомлений данного типа поле принимает значение <code>incomingCall</code>
<code>instanceData</code>	object	Объект данных с информацией об инстансе
<code>status</code>	string	Статус входящего звонка, принимает значения: <ul style="list-style-type: none"> <code>offer</code> - входящий звонок <code>pickUp</code> - отвеченный входящий звонок <code>hangUp</code> - получатель входящего звонка не взял трубку и сбросил звонок, на телефоне активирована функция "Не беспокоить" <code>missed</code> - инициатор звонка отменил звонок <code>declined</code> - пропущенный звонок
<code>timestamp</code>	integer	Время наступления события в UNIX-формате
<code>idMessage</code>	string	Идентификатор входящего звонка

Поля объекта `instanceData`

Параметр	Тип	Описание
<code>idInstance</code>	integer	Идентификатор инстанса. Размер <code>integer</code> равен <code>int64</code> . Значения лежат в диапазоне от 1 до 10 разрядов
<code>wid</code>	string	Идентификатор аккаунта в формате WhatsApp
<code>typeInstance</code>	string	Тип мессенджера для инстанса

"Статусы звонков"

****При входящем звонке система формирует по очереди два уведомления:****

- * Первое уведомление со статусом ``offer``, что указывает на получение нового входящего звонка.
- * Второе уведомление со статусом ``pickUp``, ``hangUp``, ``missed`` или ``declined``, в зависимости от исхода звонка.

****Сценарий получения уведомлений:****

1. ``offer`` + ``pickUp`` - успешный входящий звонок, адресат ответил на звонок
2. ``offer`` + ``hangUp`` - не отвеченный входящий звонок, адресат прервал соединение или на телефоне активирована функция "Не беспокоить" (положили трубку)
3. ``offer`` + ``missed`` - пропущенный входящий звонок, инициатор прервал соединение (завершил звонок звонивший)
4. ``offer`` + ``declined`` - пропущенный входящий звонок, приложение завершило соединение из-за таймаута (WhatsApp завершил звонок)

Пример тела уведомления

```
{
  "from": "79001234500@c.us",
```

```
"typeWebhook": "incomingCall",
"instanceData": {
  "idInstance": 1101123456,
  "wid": "11001234567@ec.us",
  "typeInstance": "whatsapp"
},
"status": "pickUp",
"timestamp": 1617691757,
"idMessage": "104179ED87F5328988D8834107EEBE50"
}
```

Входящая блокировка

"Уведомление временно не работает"

Входящее уведомление данного типа содержит данные о добавлении чата в список заблокированных контактов. Данное уведомление показывает факт события блокировки.

При получении блокировки от собеседника сообщения на номер будут иметь статус `sent`. Подробнее [почему статус сообщения в значении sent?](#)

Уведомление Формат уведомления

Параметр	Тип	Описание
<code>typeWebhook</code>	string	Тип входящего уведомления. Для уведомлений данного типа поле принимает значение <code>incomingBlock</code>
<code>instanceData</code>	object	Данные об инстансе
<code>timestamp</code>	integer	Время наступления события в UNIX-формате
<code>chatId</code>	integer	Идентификатор чата в формате WhatsApp. По умолчанию принимает значение <code>10000000000@c.us</code>
<code>chatState</code>	string	Состояние чата. Принимает значения: <code>blocked</code> - Контакт заблокировал чат

Поля объекта `instanceData`

Параметр	Тип	Описание
<code>idInstance</code>	integer	Идентификатор инстанса. Размер <code>integer</code> равен <code>int64</code> . Значения лежат в диапазоне от 1 до 10 разрядов
<code>wid</code>	string	Идентификатор аккаунта в формате WhatsApp
<code>typeInstance</code>	string	Тип мессенджера для инстанса

Пример тела уведомления

```
{
  "typeWebhook": "incomingBlock",
  "instanceData": {
    "idInstance": 1234,
    "wid": "11001234567@c.us",
    "typeInstance": "whatsapp"
  },
  "timestamp": 1586700690,
  "chatId": "10000000000@c.us",
  "chatState": "blocked"
}
```

Превышены ограничения на тарифе

Входящее уведомление `quotaExceeded` содержит данные о превышении ограничений чатов на тарифе Разработчик

Уведомление Формат уведомления

Параметр	Тип	Описание
<code>typeWebhook</code>	string	Тип входящего уведомления. Для уведомлений данного типа поле принимает значение <code>quotaExceeded</code>
<code>instanceData</code>	object	Данные об инстансе
<code>timestamp</code>	integer	Время наступления события в UNIX-формате
<code>quotaData</code>	object	Данные о превышении ограничений

Поля объекта `instanceData`

Параметр	Тип	Описание
<code>idInstance</code>	integer	Идентификатор инстанса. Размер <code>integer</code> равен <code>int64</code> . Значения лежат в диапазоне от 1 до 10 разрядов
<code>wid</code>	string	Идентификатор аккаунта в формате WhatsApp
<code>typeInstance</code>	string	Тип мессенджера для инстанса

Поля объекта `quotaData`

Параметр	Тип	Описание
<code>method</code>	string	Название метода или ограничения. По умолчанию принимает значение <code>correspondents</code>
<code>used</code>	string	Количество использованных чатов за месяц
<code>total</code>	string	Общее количество чатов, которых можно использовать в месяц. По умолчанию принимает значение 3
<code>status</code>	string	Статус ограничения
<code>description</code>	string	Описание превышения ограничения

Пример тела уведомления

```
{
  "typeWebhook": "quotaExceeded",
  "instanceData": {
    "idInstance": 1000100010,
    "wid": "10001234560@c.us",
    "typeInstance": "whatsapp"
  },
  "quotaData": {
    "method": "correspondents", // Ограничение чатов
    "used": 3,
    "total": 3,
    "status": "CORRESPONDENTS_QUOTA_EXCEEDED", // Превышена количество ограничений чатов
    "description": "Monthly quota has been exceeded. You can only send or receive messages from following chats: 10001234560@c.us, 10001234561@c.us, 10001234562@c.us. Please go to your personal account and change the tariff to business https://console.green-api.com"
  }
}
```


ОБЪЕКТЫ

Типы входящих уведомлений

Входящие уведомления могут быть следующих типов:

- [incomingMessageReceived](#) - уведомление о входящих сообщениях и файлах
- [outgoingMessageReceived](#) - уведомление о сообщении, отправленного с телефона
- [outgoingAPIMessageReceived](#) - уведомление о сообщении, отправленного из API
- [outgoingMessageStatus](#) - уведомление о статусах отправки/доставки/прочтении исходящих сообщений
- [stateInstanceChanged](#) - уведомление об изменении состояния авторизации инстанса
- [statusInstanceChanged](#) - уведомление об изменении состояния сокета инстанса
- [deviceInfo](#) - уведомление об устройстве (телефоне) и уровне заряда батареи
- [incomingCall](#) - уведомление о входящем звонке
- [incomingBlock](#) - уведомление о добавлении чата в список заблокированных контактов

Получение уведомлений разных типов

Для включения или отключения уведомлений по типам используйте метод [SetSettings](#)

Пример тела запроса метода [SetSettings](#)

```
{
  "outgoingWebhook": "yes",
  "outgoingMessageWebhook": "yes",
  "stateWebhook": "yes",
  "incomingWebhook": "yes",
  "deviceWebhook": "no",
  "statusInstanceChangedWebhook": "yes",
  "incomingBlockWebhook": "yes"
}
```

Получение файлов

DOWNLOADFILE



Метод предназначен для скачивания принятых и отправленных файлов. Ссылки на принятые файлы передаются во [Входящих сообщениях](#), а также их можно получить методом [LastIncomingMessages](#). Ссылки на отправленные файлы можно получить методом [LastOutgoingMessages](#).

Срок хранения файлов и соответственно возможность их первоначального скачивания ограничено временем, которое хранит их у себя WhatsApp

Запрос

POST

```
{{apiUrl}}/waInstance{{idInstance}}/downloadFile/{{apiTokenInstance}}
```

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>chatId</code>	string	Да	Идентификатор личного или группового чата историю сообщений которого требуется получить
<code>idMessage</code>	string	Да	Идентификатор сообщения

Ответ

Поля ответа

Поле	Тип	Описание
<code>downloadUrl</code>	string	Ссылка на файл из сообщения

Пример тела ответа

```
{
  "downloadUrl": "https://sw-media.storage.yandexcloud.net/1103912412/a6679d42-2f7f-4121-acfe-1f993dfcf123.png"
}
```

Ошибки DownloadFile

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
400	Bad Request Validation failed	Ошибка валидации
400	Validation failed. Details: 'value' must have at least 2 keys	Ошибка валидации. Поле <code>value</code> должно содержать минимум 2 символа
400	File message encrypted url not found by chatId 790000312312@c.us and idMessage A322F800D3F12CD4858CC947DAFB77A2	Файл отсутствует в сообщении
500	Internal error when downloading file by chatId 790000312312@c.us and idMessage A322F800D3F12CD4858CC947DAFB77A2	Файл не доступен на серверах WhatsApp, загрузка через API невозможна

1.3.5 Журналы

Журналы

В данном разделе представлены методы по работе с журналами входящих и отправленных сообщений

- [Получить историю сообщений чата](#)
- [Получить сообщение чата](#)
- [Получить журнал входящих сообщений](#)
- [Получить журнал отправленных сообщений](#)

GetChatHistory



Метод возвращает историю сообщений чата.

Чтобы получить **входящие сообщения**, включите настройку `Получать уведомления о входящих сообщениях и файлах` методом `SetSettings` или через [личный кабинет](#).

Чтобы получить **статусы отправленных сообщений**, включите настройку `Получать уведомления о статусах отправленных сообщений` методом `SetSettings` или через [личный кабинет](#).

Применение настроек занимает до 5 минут.

Чтобы получить сообщения, отправленные и полученные с телефона **до подключения инстанса**, необходимо после сохранения настроек на инстансе сделать [разлогин](#) и [авторизоваться](#) повторно.

Метод `GetChatHistory` возвращает только те сообщения, которые системе отдает WhatsApp.

Появление сообщений в журнале может занимать время до 2 минут, методы журналов необходимо использовать только для получения истории переписки. Для быстрого получения сообщений необходимо использовать [систему уведомлений](#).

ЗАПРОС

Для получения истории сообщений требуется выполнить запрос по адресу:

POST

```
{{apiUrl}}/waInstance{{idInstance}}/getChatHistory/{{apiTokenInstance}}
```

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>chatId</code>	string	Да	Идентификатор личного или группового чата историю сообщений которого требуется получить
<code>count</code>	integer	Нет	Количество сообщений для получения. Значение по умолчанию <code>100</code>

Пример тела запроса

Запрос последних 10 сообщений:

```
{
  "chatId": "11001234567@c.us",
  "count": 10
}
```

ОТВЕТ

Ответ содержит список всех полученных и отправленных сообщений в чате. Сортировка по убыванию даты отправки сообщения.

Поля ответа

Массив объектов с полями:

Поле	Тип	Описание
type	string	Вид сообщения: <code>outgoing</code> - исходящее сообщение; <code>incoming</code> - входящее сообщение
idMessage	string	Идентификатор входящего сообщения
timestamp	integer	Время принятия сообщения в UNIX-формате
statusMessage	string	Статус исходящего сообщения. Присутствует только для <code>type = outgoing</code> . Возможные значения:
		<code>noAccount</code> - нет аккаунта WhatsApp на номере телефона
		<code>notInGroup</code> - не состоите в данной группе
		<code>pending</code> - сообщение отправляется
		<code>sent</code> - отправлено
		<code>delivered</code> - доставлено
		<code>read</code> - прочитано/просмотрено/прослушано
		<code>yellowCard</code> - приостановка отправки сообщений из-за спамерской активности. Сообщения отправленные после получения статуса хранятся в очереди к отправке 24 часа. Для продолжения работы инстанса требуется сделать перезагрузку инстанса
sendByApi	boolean	Флаг указывает отправлено сообщение из API или нет. Присутствует только для <code>type = outgoing</code>
typeMessage	string	Тип сообщения, возможные значения:
		<code>textMessage</code> - текстовое сообщение
		<code>imageMessage</code> - сообщение с изображением
		<code>videoMessage</code> - видео сообщение
		<code>documentMessage</code> - сообщение с файлом документа
		<code>audioMessage</code> - аудио сообщение
		<code>stickerMessage</code> - сообщение со стикером
		<code>reactionMessage</code> - сообщение с реакцией
		<code>locationMessage</code> - сообщение геолокации
		<code>contactMessage</code> - сообщение с контактом
		<code>extendedTextMessage</code> - сообщение со ссылкой и превью
		<code>pollMessage</code> - сообщение с опросом
		<code>pollUpdateMessage</code> - сообщение с обновлением опросом. Присутствует только для <code>type = incoming</code>
		<code>quotedMessage</code> - сообщение с цитированием
chatId	string	Идентификатор чата , в котором получено сообщение.
senderId	string	Идентификатор отправителя сообщения. Присутствует только для <code>type = incoming</code>
senderName	string	Имя отправителя сообщения. Присутствует только для <code>type = incoming</code>
senderContactName	string	

Поле	Тип	Описание
		Имя отправителя из списка контактов текущего аккаунта. Присутствует только для <code>type = incoming</code>
<code>isForwarded</code>	boolean	Флаг указывает является ли сообщение пересланным
<code>forwardingScore</code>	integer	Количество пересылок сообщения
<code>textMessage</code>	string	Текст сообщения, если <code>typeMessage = textMessage / extendedTextMessage</code>
<code>downloadUrl</code>	string	Ссылка на скачивание файла, если <code>typeMessage = imageMessage / videoMessage / documentMessage / audioMessage / stickerMessage</code>
<code>caption</code>	string	Описание файла, если <code>typeMessage = imageMessage / videoMessage / documentMessage / audioMessage / stickerMessage</code>
<code>fileName</code>	string	Название файла, если <code>typeMessage = imageMessage / videoMessage / documentMessage / audioMessage / stickerMessage</code> . Поле заполняется корректно только для <code>typeMessage = documentMessage</code> , в остальных случаях название файла генерируется автоматически
<code>jpegThumbnail</code>	string	Превью изображения в base64 кодировке, если <code>typeMessage = imageMessage / videoMessage / documentMessage / audioMessage / stickerMessage</code>
<code>mimeType</code>	string	Тип файла, согласно классификации Media Types , если <code>typeMessage = imageMessage / videoMessage / documentMessage / audioMessage / stickerMessage</code>
<code>isAnimated</code>	boolean	Флаг указывает анимирован ли файл, если <code>typeMessage = imageMessage / videoMessage / documentMessage / audioMessage / stickerMessage</code>
<code>location</code>	object	Объект о структуре локации, если <code>typeMessage = locationMessage</code>
<code>contact</code>	object	Объект о структуре контакта, если <code>typeMessage = contactMessage</code>
<code>extendedTextMessage</code>	object	Объект с текстовым сообщением (расширенный), если <code>typeMessage = extendedTextMessage / quotedMessage</code>
<code>extendedTextMessageData</code>	object	Объект с реакцией, если <code>typeMessage = reactionMessage</code>
<code>pollMessageData</code>	object	Объект о структуре данных опроса, если <code>typeMessage = pollMessage / pollUpdateMessage</code>
<code>quotedMessage</code>	object	Объект данных о цитируемом сообщении, если <code>typeMessage = quotedMessage / reactionMessage</code>

Поля объекта `location`:

Поле	Тип	Описание
<code>nameLocation</code>	string	Название локации
<code>address</code>	string	Адрес локации
<code>latitude</code>	double	Широта локации
<code>longitude</code>	double	Долгота локации
<code>jpegThumbnail</code>	string	Превью изображения в base64 кодировке
<code>isForwarded</code>	boolean	Флаг указывает является ли сообщение пересланным
<code>forwardingScore</code>	integer	Количество пересылок сообщения

Поля объекта `contact` :

Поле	Тип	Описание
<code>displayName</code>	string	Отображаемое имя контакта
<code>vcard</code>	string	Структура VCard (визитной карточки контакта)
<code>isForwarded</code>	boolean	Флаг указывает является ли сообщение пересланным
<code>forwardingScore</code>	integer	Количество пересылок сообщения

Поля объекта `extendedTextMessage` :

Поле	Тип	Описание
<code>text</code>	string	Текст ссылки
<code>description</code>	string	Описание ссылки
<code>title</code>	string	Заголовок ссылки
<code>previewType</code>	string	Тип превью ссылки. Значение <code>previewType: None</code> устарело
<code>jpegThumbnail</code>	string	Превью изображения в base64 кодировке
<code>stanzaId</code>	string	ID цитируемого сообщения. Присутствует только для <code>typeMessage = quotedMessage</code>
<code>participant</code>	string	ID чата получателя/отправителя. Присутствует только для <code>typeMessage = quotedMessage</code>
<code>isForwarded</code>	boolean	Флаг указывает является ли сообщение пересланным
<code>forwardingScore</code>	integer	Количество пересылок сообщения

Поля объекта `extendedTextMessageData` :

Поле	Тип	Описание
<code>text</code>	string	Реакция (эмоджи) на сообщение

Поля объекта `pollMessageData` :

Поле	Тип	Описание
<code>stanzaId</code>	string	id сообщения с исходящим опросом. Присутствует только для <code>typeMessage = pollUpdateMessage</code>
<code>name</code>	string	Текст опроса
<code>options</code>	array	Массив данных о вариантах выбора. Присутствует только для <code>typeMessage = pollMessage</code>
<code>votes</code>	array	Массив данных ответов. Присутствует только для <code>typeMessage = pollUpdateMessage</code>
<code>multipleAnswers</code>	boolean	Флаг о разрешении нескольких ответов

Поля массива `options`

Параметр	Тип	Описание
<code>optionName</code>	string	Текст варианта выбора

Поля массива votes :

Параметр	Тип	Описание
optionName	string	Текст варианта выбора
optionVoters	array	Массив идентификаторов проголосовавших аккаунтов

Поля объекта quotedMessage :

Поле	Тип	Описание
stanzaId	string	ID цитируемого сообщения
participant	string	ID чата получателя
typeMessage	string	Тип цитируемого сообщения

Остальные поля объекта quotedMessage заполняются в зависимости от типа цитируемого сообщения и идентичны полям описанных выше

Пример тела ответа

```
[
  {
    "type": "incoming",
    "idMessage": "90B14F14A253D33F4A9CD84123456789",
    "timestamp": 170652263,
    "typeMessage": "textMessage",
    "chatId": "120363153000000000@g.us",
    "textMessage": "Привет",
    "senderId": "7000000012@c.us",
    "senderName": "Николай",
    "senderContactName": "Коля"
  },
  {
    "type": "outgoing",
    "idMessage": "BAE5143000000000",
    "timestamp": 1706761225,
    "typeMessage": "textMessage",
    "chatId": "7000000012@c.us",
    "textMessage": "Привет",
    "statusMessage": "read",
    "sendByApi": true
  },
  {
    "type": "incoming",
    "idMessage": "EAAE5A5CD000000000",
    "timestamp": 1706601620,
    "typeMessage": "imageMessage", //"/videoMessage"/"documentMessage"/"audioMessage"/"stickerMessage"
    "chatId": "120363153000000000@g.us",
    "downloadUrl": "https://api.green-api.com/waInstance1234/downloadFile/EAI1BD1AE042DC4F3609867126309067C.jpg",
    "caption": "Как тебе?",
    "fileName": "03963bac-6020-423c-80df-1aa123456789.jpg",
    "jpegThumbnail": "/9j/4AAQSkZJ/aAAgBAwEPwCVYw/xAApEAAcAx/2k0Vh/VKAoAH/LeFhX/+AAMA/9k=",
    "mimeType": "image/jpeg",
    "isAnimated": false,
    "isForwarded": true,
    "forwardingScore": 1,
    "senderId": "7000000012@c.us",
    "senderName": "Николай",
    "senderContactName": "Коля"
  },
  {
    "type": "outgoing",
    "idMessage": "3A1EBA0F23004BD415BB",
    "timestamp": 1706601621,
    "typeMessage": "reactionMessage",
    "chatId": "7000000012@c.us",
    "extendedTextMessageData": {
      "text": "😄"
    },
    "quotedMessage": {
      "stanzaId": "B4D7D8A1A7D887D5580F0000000000",
      "participant": "7000000012@c.us",
    },
    "statusMessage": "read",
    "sendByApi": false
  },
  {
    "type": "outgoing",
    "idMessage": "BAE5870000000000",
    "timestamp": 1706761414,
```

```

"typeMessage": "LocationMessage",
"chatId": "70000000012@c.us",
"location": {
  "nameLocation": "Cafe",
  "address": "Астана",
  "jpegThumbnail": "/9j/4AAQSkZJRgABAQAAASABIAAD/4QBVRXh",
  "latitude": 51.12800000000000,
  "longitude": 71.43076200000000,
  "forwardingScore": 0,
  "isForwarded": false
},
"statusMessage": "read",
"sendByApi": true
},
{
  "type": "incoming",
  "idMessage": "3AA2C1D8356000000000",
  "timestamp": 1706675513,
  "typeMessage": "contactMessage",
  "chatId": "70000000012@c.us",
  "contact": {
    "displayName": "Виктор Петров",
    "vcard": "BEGIN:VCARD\nVERSION:3.0\nN:;Виктор Петров;;;\nFN:Виктор Петров\nTEL;type=CELL;type=VOICE;waid=70000000013:+7 000 000-00-13\nEND:VCARD",
    "forwardingScore": 0,
    "isForwarded": false
  },
  "senderId": "70000000012@c.us",
  "senderName": "Николай",
  "senderContactName": "Коля"
},
{
  "type": "incoming",
  "idMessage": "3A66D7F0B50000000000",
  "timestamp": 1706675928,
  "typeMessage": "extendedTextMessage",
  "chatId": "70000000012@c.us",
  "textMessage": "Hqhghqh\nhttps://green-api.com/",
  "extendedTextMessage": {
    "text": "Hqhghqh\nhttps://green-api.com/",
    "description": "GREEN API - интеграция с WhatsApp на любом языке программирования PHP, JavaScript, 1C, Python, Java, C#, VBA",
    "title": "Green API для отправки и получения сообщений WhatsApp",
    "previewType": "None",
    "jpegThumbnail": "/9j/4AAQSkZJRgABAQAAASAB9k=",
    "forwardingScore": 0,
    "isForwarded": false
  },
  "senderId": "70000000012@c.us",
  "senderName": "Николай",
  "senderContactName": "Коля"
},
{
  "type": "outgoing",
  "idMessage": "460D84E00000000000",
  "timestamp": 1706762412,
  "typeMessage": "pollMessage",
  "chatId": "70000000012@c.us",
  "pollMessageData": {
    "name": "Выберите любимый цвет",
    "options": [
      {
        "optionName": "Красный"
      },
      {
        "optionName": "Зеленый"
      }
    ],
    "multipleAnswers": true
  },
  "statusMessage": "delivered",
  "sendByApi": true
},
{
  "type": "incoming",
  "idMessage": "3A0493ED900000000000",
  "timestamp": 1706676447,
  "typeMessage": "pollUpdateMessage",
  "chatId": "70000000012@c.us",
  "pollMessageData": {
    "name": "Выберите любимый цвет",
    "stanzaId": "460D84E00000000000",
    "votes": [
      {
        "optionName": "Красный",
        "optionVoters": [
          "70000000012@c.us"
        ]
      },
      {
        "optionName": "Зеленый",
        "optionVoters": []
      }
    ],
    "multipleAnswers": true
  },

```

```

"senderId": "7000000012@с.us",
"senderName": "Николай",
"senderContactName": "Коля"
},
{
  "type": "outgoing",
  "idMessage": "B1494991277FD00000000000000000000",
  "timestamp": 1706849516,
  "typeMessage": "quotedMessage",
  "chatId": "7000000012@с.us",
  "extendedTextMessage": {
    "text": "Черный",
    "stanzaId": "460D84E00000000000",
    "participant": "7000000012@с.us"
  },
  "quotedMessage": {
    "stanzaId": "460D84E00000000000",
    "participant": "7000000012@с.us",
    "typeMessage": "pollMessage",
    "pollMessageData": {
      "name": "Выберите любимый цвет",
      "options": [],
      "multipleAnswers": true
    }
  }
},
"statusMessage": "delivered",
"sendByApi": false
},
{
  "type": "outgoing",
  "idMessage": "21D6C50B400DBCEC3000000000000000",
  "timestamp": 1706852882,
  "typeMessage": "quotedMessage",
  "chatId": "7000000012@с.us",
  "extendedTextMessage": {
    "text": "Кrypto",
    "stanzaId": "79DC2458992097F00000000000000000",
    "participant": "7000000012@с.us"
  },
  "quotedMessage": {
    "stanzaId": "79DC2458992097F00000000000000000",
    "participant": "7000000012@с.us",
    "typeMessage": "imageMessage", /*"videoMessage"/"documentMessage"/"audioMessage"/"stickerMessage"
    "downloadUrl": "",
    "caption": "",
    "fileName": "",
    "jpegThumbnail": "/9j/4AAQSkZJRgABAQAAQABAAD/2wCEAgQ70/wD/2Q==",
    "mimeType": "image/jpeg",
    "isAnimated": false,
    "isForwarded": false,
    "forwardingScore": 0
  }
},
"statusMessage": "read",
"sendByApi": true
}
]

```

Ошибки GetChatHistory

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
400	Bad Request Validation failed	Ошибка валидации
400	Bad Request Validation failed. Details: 'count' must be a number	Некорректный тип данных для поля count либо значение поля превышает максимальное безопасное целочисленное значение

GetMessage



Метод возвращает сообщение чата.

Чтобы получить **входящее сообщение**, включите настройку `Получать уведомления о входящих сообщениях и файлах` методом `SetSettings` или через [личный кабинет](#).

Чтобы получить **статус отправленного сообщения**, включите настройку `Получать уведомления о статусах отправленных сообщений` методом `SetSettings` или через [личный кабинет](#).

Применение настроек занимает до 5 минут.

Чтобы получить сообщения, отправленные и полученные с телефона **до подключения инстанса**, необходимо после сохранения настроек на инстансе сделать [разлогин](#) и [авторизоваться](#) повторно.

Метод GetMessage возвращает только те сообщения, которые системе отдает WhatsApp.

Появление сообщений в журнале может занимать время до 2 минут, методы журналов необходимо использовать только для получения истории переписки. Для быстрого получения сообщений необходимо использовать [систему уведомлений](#).

ЗАПРОС

Для получения сообщения требуется выполнить запрос по адресу:

```
POST {{apiUrl}}/waInstance{{idInstance}}/getMessage/{{apiTokenInstance}}
```

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>chatId</code>	string	Да	Идентификатор личного или группового чата, сообщение которого требуется получить
<code>idMessage</code>	string	Да	ID сообщения

Пример тела запроса

Запрос сообщения:

```
{
  "chatId": "120363043968066561@g.us",
  "idMessage": "BAE5F4886F6F2D05"
}
```

ОТВЕТ

Ответ содержит полученное или отправленное сообщение из чата.

Поля ответа

Объект с полями:

Поле	Тип	Описание
type	string	Вид сообщения: <code>outgoing</code> - исходящее сообщение; <code>incoming</code> - входящее сообщение
idMessage	string	Идентификатор входящего сообщения
timestamp	integer	Время принятия сообщения в UNIX-формате
statusMessage	string	Статус исходящего сообщения. Присутствует только для <code>type = outgoing</code> . Возможные значения:
		<code>noAccount</code> - нет аккаунта WhatsApp на номере телефона
		<code>notInGroup</code> - не состоите в данной группе
		<code>pending</code> - сообщение отправляется
		<code>sent</code> - отправлено
		<code>delivered</code> - доставлено
		<code>read</code> - прочитано/просмотрено/прослушано
		<code>yellowCard</code> - приостановка отправки сообщений из-за спамерской активности. Сообщения отправленные после получения статуса хранятся в очереди к отправке 24 часа. Для продолжения работы инстанса требуется сделать перезагрузку инстанса
sendByApi	boolean	Флаг указывает отправлено сообщение из API или нет. Присутствует только для <code>type = outgoing</code>
typeMessage	string	Тип сообщения, возможные значения:
		<code>textMessage</code> - текстовое сообщение
		<code>imageMessage</code> - сообщение с изображением
		<code>videoMessage</code> - видео сообщение
		<code>documentMessage</code> - сообщение с файлом документа
		<code>audioMessage</code> - аудио сообщение
		<code>stickerMessage</code> - сообщение со стикером
		<code>reactionMessage</code> - сообщение с реакцией
		<code>locationMessage</code> - сообщение геолокации
		<code>contactMessage</code> - сообщение с контактом
		<code>extendedTextMessage</code> - сообщение со ссылкой и превью
		<code>pollMessage</code> - сообщение с опросом
		<code>pollUpdateMessage</code> - сообщение с обновлением опросом. Присутствует только для <code>type = incoming</code>
		<code>quotedMessage</code> - сообщение с цитированием
chatId	string	Идентификатор чата , в котором получено сообщение.
senderId	string	Идентификатор отправителя сообщения. Присутствует только для <code>type = incoming</code>
senderName	string	Имя отправителя сообщения. Присутствует только для <code>type = incoming</code>
senderContactName	string	

Поле	Тип	Описание
		Имя отправителя из списка контактов текущего аккаунта. Присутствует только для <code>type = incoming</code>
<code>isForwarded</code>	boolean	Флаг указывает является ли сообщение пересланным
<code>forwardingScore</code>	integer	Количество пересылок сообщения
<code>textMessage</code>	string	Текст сообщения, если <code>typeMessage = textMessage / extendedTextMessage</code>
<code>downloadUrl</code>	string	Ссылка на скачивание файла, если <code>typeMessage = imageMessage / videoMessage / documentMessage / audioMessage / stickerMessage</code>
<code>caption</code>	string	Описание файла, если <code>typeMessage = imageMessage / videoMessage / documentMessage / audioMessage / stickerMessage</code>
<code>fileName</code>	string	Название файла, если <code>typeMessage = imageMessage / videoMessage / documentMessage / audioMessage / stickerMessage</code> . Поле заполняется корректно только для <code>typeMessage = documentMessage</code> , в остальных случаях название файла генерируется автоматически
<code>jpegThumbnail</code>	string	Превью изображения в base64 кодировке, если <code>typeMessage = imageMessage / videoMessage / documentMessage / audioMessage / stickerMessage</code>
<code>mimeType</code>	string	Тип файла, согласно классификации Media Types , если <code>typeMessage = imageMessage / videoMessage / documentMessage / audioMessage / stickerMessage</code>
<code>isAnimated</code>	boolean	Флаг указывает анимирован ли файл, если <code>typeMessage = imageMessage / videoMessage / documentMessage / audioMessage / stickerMessage</code>
<code>location</code>	object	Объект о структуре локации, если <code>typeMessage = locationMessage</code>
<code>contact</code>	object	Объект о структуре контакта, если <code>typeMessage = contactMessage</code>
<code>extendedTextMessage</code>	object	Объект с текстовым сообщением (расширенный), если <code>typeMessage = extendedTextMessage / quotedMessage</code>
<code>extendedTextMessageData</code>	object	Объект с реакцией, если <code>typeMessage = reactionMessage</code>
<code>pollMessageData</code>	object	Объект о структуре данных опроса, если <code>typeMessage = pollMessage / pollUpdateMessage</code>
<code>quotedMessage</code>	object	Объект данных о цитируемом сообщении, если <code>typeMessage = quotedMessage / reactionMessage</code>

Поля объекта `location`:

Поле	Тип	Описание
<code>nameLocation</code>	string	Название локации
<code>address</code>	string	Адрес локации
<code>latitude</code>	double	Широта локации
<code>longitude</code>	double	Долгота локации
<code>jpegThumbnail</code>	string	Превью изображения в base64 кодировке
<code>isForwarded</code>	boolean	Флаг указывает является ли сообщение пересланным
<code>forwardingScore</code>	integer	Количество пересылок сообщения

Поля объекта `contact` :

Поле	Тип	Описание
<code>displayName</code>	string	Отображаемое имя контакта
<code>vcard</code>	string	Структура VCard (визитной карточки контакта)
<code>isForwarded</code>	boolean	Флаг указывает является ли сообщение пересланным
<code>forwardingScore</code>	integer	Количество пересылок сообщения

Поля объекта `extendedTextMessage` :

Поле	Тип	Описание
<code>text</code>	string	Текст ссылки
<code>description</code>	string	Описание ссылки
<code>title</code>	string	Заголовок ссылки
<code>previewType</code>	string	Тип превью ссылки. Значение <code>previewType: None</code> устарело
<code>jpegThumbnail</code>	string	Превью изображения в base64 кодировке
<code>stanzaId</code>	string	ID цитируемого сообщения. Присутствует только для <code>typeMessage = quotedMessage</code>
<code>participant</code>	string	ID чата получателя/отправителя. Присутствует только для <code>typeMessage = quotedMessage</code>
<code>isForwarded</code>	boolean	Флаг указывает является ли сообщение пересланным
<code>forwardingScore</code>	integer	Количество пересылок сообщения

Поля объекта `extendedTextMessageData` :

Поле	Тип	Описание
<code>text</code>	string	Реакция (эмоджи) на сообщение

Поля объекта `pollMessageData` :

Поле	Тип	Описание
<code>stanzaId</code>	string	id сообщения с исходящим опросом. Присутствует только для <code>typeMessage = pollUpdateMessage</code>
<code>name</code>	string	Текст опроса
<code>options</code>	array	Массив данных о вариантах выбора. Присутствует только для <code>typeMessage = pollMessage</code>
<code>votes</code>	array	Массив данных ответов. Присутствует только для <code>typeMessage = pollUpdateMessage</code>
<code>multipleAnswers</code>	boolean	Флаг о разрешении нескольких ответов

Поля массива `options`

Параметр	Тип	Описание
<code>optionName</code>	string	Текст варианта выбора

Поля массива votes :

Параметр	Тип	Описание
optionName	string	Текст варианта выбора
optionVoters	array	Массив идентификаторов проголосовавших аккаунтов

Поля объекта quotedMessage :

Поле	Тип	Описание
stanzaId	string	ID цитируемого сообщения
participant	string	ID чата получателя
typeMessage	string	Тип цитируемого сообщения

Остальные поля объекта quotedMessage заполняются в зависимости от типа цитируемого сообщения и идентичны полям описанных выше

Пример тела ответа

Примеры тел ответа входящего сообщения Текстовое сообщение

```
{
  "type": "incoming",
  "idMessage": "9DB14F14A253D33F4A9CD84123456789",
  "timestamp": 170652263,
  "typeMessage": "textMessage",
  "chatId": "120363153000000000@eg.us",
  "textMessage": "Привет",
  "senderId": "7000000012@.us",
  "senderName": "Николай",
  "senderContactName": "Коля"
}
```

Сообщение с документом или медиа

```
{
  "type": "incoming",
  "idMessage": "EAAE5A5CD000000000",
  "timestamp": 1706601620,
  "typeMessage": "imageMessage", /*"videoMessage"/"documentMessage"/"audioMessage"/"stickerMessage"*/
  "chatId": "120363153000000000@eg.us",
  "downloadUrl": "https://api.green-api.com/waInstance1234/downloadFile/EAI1BD1AE042DC4F3609867126309067C.jpg",
  "caption": "Как тебе?",
  "fileName": "03963bac-6020-423c-80df-1aa123456789.jpg",
  "jpegThumbnail": "/9j/4AAQSkZJ/aAAgBAwEPwCVYw/xAApEAACAx/2k0Vh/VKAoAH/LeFhX/+AAMA/9k=",
  "mimeType": "image/jpeg",
  "isAnimated": false,
  "isForwarded": true,
  "forwardingScore": 1,
  "senderId": "7000000012@.us",
  "senderName": "Николай",
  "senderContactName": "Коля"
}
```

Сообщение с реакцией

```
{
  "type": "incoming",
  "idMessage": "3A1EBA0F23004BD415BB",
  "timestamp": 1706601621,
  "typeMessage": "reactionMessage",
  "chatId": "7000000012@.us",
  "extendedTextMessageData": {
    "text": "😄"
  },
  "quotedMessage": {
    "stanzaId": "B4D7D8A1A7D887D5580F000000000000",
    "participant": "7000000012@.us",
  },
  "senderId": "7000000013@.us",
  "senderName": "Николай",
  "senderContactName": "Коля"
}
```

Сообщение с геолокацией

```
{
  "type": "incoming",
  "idMessage": "3A8427EF85F000000000",
  "timestamp": 1706674611,
  "typeMessage": "locationMessage",
  "chatId": "70000000012@с.us",
  "location": {
    "nameLocation": "Cafe",
    "address": "Астана",
    "jpegThumbnail": "/9j/4AAQSkZJRgABAQAAASABIAAD/4QBYRX",
    "latitude": 51.12800000000000,
    "longitude": 71.43076200000000,
    "forwardingScore": 0,
    "isForwarded": false
  },
  "senderId": "70000000012@с.us",
  "senderName": "Николай",
  "senderContactName": "Коля"
}
```

Сообщение с контактом

```
{
  "type": "incoming",
  "idMessage": "3AA2C1D835600000000000",
  "timestamp": 1706675513,
  "typeMessage": "contactMessage",
  "chatId": "70000000012@с.us",
  "contact": {
    "displayName": "Виктор Петров",
    "vcard": "BEGIN:VCARD\nVERSION:3.0\nn:;Виктор Петров;;\nfn:Виктор Петров\nTEL;type=CELL;type=VOICE;waid=70000000013:+7 000 000-00-13\nEND:VCARD",
    "forwardingScore": 0,
    "isForwarded": false
  },
  "senderId": "70000000012@с.us",
  "senderName": "Николай",
  "senderContactName": "Коля"
}
```

Расширенное текстовое сообщение

```
{
  "type": "incoming",
  "idMessage": "3A66D7F0B5000000000000",
  "timestamp": 1706675928,
  "typeMessage": "extendedTextMessage",
  "chatId": "70000000012@с.us",
  "textMessage": "Hqhghqh\nhttps://green-api.com/",
  "extendedTextMessage": {
    "text": "Hqhghqh\nhttps://green-api.com/",
    "description": "GREEN API - интеграция с WhatsApp на любом языке программирования PHP, JavaScript, 1C, Python, Java, C#, VBA",
    "title": "Green API для отправки и получения сообщений Whatsapp",
    "previewType": "None",
    "jpegThumbnail": "/9j/4AAQSkZJRgABAQAAASAB9k=",
    "forwardingScore": 0,
    "isForwarded": false
  },
  "senderId": "70000000012@с.us",
  "senderName": "Николай",
  "senderContactName": "Коля"
}
```

Сообщение с опросом

```
{
  "type": "incoming",
  "idMessage": "3A71511C3A000000000000",
  "timestamp": 1706676278,
  "typeMessage": "pollMessage",
  "chatId": "70000000012@с.us",
  "pollMessageData": {
    "name": "Выберите любимый цвет",
    "options": [
      {
        "optionName": "Красный"
      },
      {
        "optionName": "Зеленый"
      }
    ]
  },
  "multipleAnswers": true
},
  "senderId": "70000000012@с.us",
  "senderName": "Николай",
  "senderContactName": "Коля"
}
```

Сообщение с обновлением опроса

```

{
  "type": "incoming",
  "idMessage": "3A0493ED9000000000000",
  "timestamp": 1706676447,
  "typeMessage": "pollUpdateMessage",
  "chatId": "70000000012@с.us",
  "pollMessageData": {
    "name": "Куда пойдём?",
    "stanzaId": "46618B9883B9DF50F123456789123456",
    "votes": [
      {
        "optionName": "Пляж",
        "optionVoters": [
          "70000000012@с.us"
        ]
      },
      {
        "optionName": "Горы",
        "optionVoters": []
      }
    ],
    "multipleAnswers": true
  },
  "senderId": "70000000012@с.us",
  "senderName": "Николай",
  "senderContactName": "Коля"
}

```

Цитированное сообщение

```

{
  "type": "incoming",
  "idMessage": "3AB0E938BA00000000000",
  "timestamp": 1706677013,
  "typeMessage": "quotedMessage",
  "chatId": "70000000012@с.us",
  "extendedTextMessage": {
    "text": "За пляж",
    "stanzaId": "46618B9883B9DF50F123456789123456",
    "participant": "70000000012@с.us"
  },
  "quotedMessage": {
    "stanzaId": "46618B9883B9DF50F123456789123456",
    "participant": "70000000012@с.us",
    "typeMessage": "pollMessage",
    "pollMessageData": {
      "name": "Куда пойдём?",
      "options": [],
      "multipleAnswers": true
    }
  },
  "senderId": "70000000012@с.us",
  "senderName": "Николай",
  "senderContactName": "Коля"
}

```

```

{
  "type": "incoming",
  "idMessage": "3A70B27A6020000000000",
  "timestamp": 1706677254,
  "typeMessage": "quotedMessage",
  "chatId": "70000000012@с.us",
  "extendedTextMessage": {
    "text": "Круто!",
    "stanzaId": "EAAE5A5CD0000000000",
    "participant": "70000000012@с.us"
  },
  "quotedMessage": {
    "stanzaId": "EAAE5A5CD0000000000",
    "participant": "70000000012@с.us",
    "typeMessage": "imageMessage", //"/videoMessage"/'documentMessage'/'audioMessage'/'stickerMessage'
    "downloadUrl": "",
    "caption": "",
    "fileName": "",
    "jpegThumbnail": "/9j/4AAQSkZJRgABAQAA=",
    "mimeType": "image/jpeg",
    "isAnimated": false,
    "isForwarded": false,
    "forwardingScore": 0
  },
  "senderId": "70000000012@с.us",
  "senderName": "Николай",
  "senderContactName": "Коля"
}

```

Примеры тел ответа исходящего сообщения Текстовое сообщение

```
{
  "type": "outgoing",
  "idMessage": "BAE5143000000000",
  "timestamp": 1706761225,
  "typeMessage": "textMessage",
  "chatId": "70000000012@ec.us",
  "textMessage": "Привет",
  "statusMessage": "read",
  "sendByApi": true
}
```

Медиа и файловые сообщения

```
{
  "type": "outgoing",
  "idMessage": "BAE5F94000000000",
  "timestamp": 1706765613,
  "typeMessage": "imageMessage", // "videoMessage" / "documentMessage" / "audioMessage" / "stickerMessage"
  "chatId": "70000000012@ec.us",
  "downloadUrl": "https://api.green-api.com/waInstance1234/downloadFile/3EB0BDDC948FDFB3D4FA.jpg",
  "caption": "Как тебе?",
  "fileName": "b89ded5e-1a3b-4168-895a-540123456789.jpg",
  "jpegThumbnail": "/9j/4AAQSkZJRgABAQAAQABAAQ/FFFCF/Z",
  "mimeType": "image/jpeg",
  "isAnimated": false,
  "isForwarded": true,
  "forwardingScore": 2,
  "statusMessage": "read",
  "sendByApi": false
}
```

Сообщение с реакцией

```
{
  "type": "outgoing",
  "idMessage": "3A1EBA0F23004BD415B8",
  "timestamp": 1706601621,
  "typeMessage": "reactionMessage",
  "chatId": "70000000012@ec.us",
  "extendedTextMessageData": {
    "text": "😄"
  },
  "quotedMessage": {
    "stanzaId": "B4D7D8A1A7D887D5580F000000000000",
    "participant": "70000000012@ec.us"
  },
  "statusMessage": "read",
  "sendByApi": false
}
```

Сообщение с геолокацией

```
{
  "type": "outgoing",
  "idMessage": "BAE5870000000000",
  "timestamp": 1706761414,
  "typeMessage": "locationMessage",
  "chatId": "70000000012@ec.us",
  "location": {
    "nameLocation": "Cafe",
    "address": "Астана",
    "jpegThumbnail": "/9j/4AAQSkZJRgABAQAAQABIAAD/4QBVRX",
    "latitude": 51.12800000000000,
    "longitude": 71.43076200000000,
    "forwardingScore": 0,
    "isForwarded": false
  },
  "statusMessage": "read",
  "sendByApi": true
}
```

Сообщение с контактом

```
{
  "type": "outgoing",
  "idMessage": "F39F2DCE0000000000",
  "timestamp": 1706762400,
  "typeMessage": "contactMessage",
  "chatId": "70000000012@ec.us",
  "contact": {
    "displayName": "Виктор Петров",
    "vcard": "BEGIN:VCARD\nVERSION:3.0\nn:;Виктор Петров;;;\nfn:Виктор Петров\nTEL;type=CELL;type=VOICE;waid=70000000013:+7 000 000-00-13\nEND:VCARD",
    "forwardingScore": 0,
    "isForwarded": false
  },
  "statusMessage": "read",
}
```

```

    "sendByApi": false
  }

```

Расширенное текстовое сообщение

```

{
  "type": "outgoing",
  "idMessage": "BAE56E0000000000",
  "timestamp": 1706849285,
  "typeMessage": "extendedTextMessage",
  "chatId": "7000000012@c.us",
  "textMessage": "https://green-api.com/",
  "extendedTextMessage": {
    "text": "https://green-api.com/",
    "description": "GREEN API - интеграция с WhatsApp на любом языке программирования PHP, JavaScript, 1C, Python, Java, C#, VBA",
    "title": "Green API для отправки и получения сообщений Whatsapp",
    "previewType": "None",
    "jpegThumbnail": "/9j/2wbDAAAYEBQVFBAYGbQYhBwYICh/SIAMkIQnxOEIQkIEIQkIEIQkIEIQkI//9k=",
    "forwardingScore": 0,
    "isForwarded": false
  },
  "statusMessage": "sent",
  "sendByApi": true
}

```

Сообщение с опросом

```

{
  "type": "outgoing",
  "idMessage": "460D84E00000000000",
  "timestamp": 1706762412,
  "typeMessage": "pollMessage",
  "chatId": "7000000012@c.us",
  "pollMessageData": {
    "name": "Выберите любимый цвет",
    "options": [
      {
        "optionName": "Красный"
      },
      {
        "optionName": "Зеленый"
      }
    ],
    "multipleAnswers": true
  },
  "statusMessage": "delivered",
  "sendByApi": true
}

```

Цитированное сообщение

```

{
  "type": "outgoing",
  "idMessage": "B1494991277FD00000000000000000000",
  "timestamp": 1706849516,
  "typeMessage": "quotedMessage",
  "chatId": "7000000012@c.us",
  "extendedTextMessage": {
    "text": "Черный",
    "stanzaId": "460D84E00000000000",
    "participant": "7000000012@c.us"
  },
  "quotedMessage": {
    "stanzaId": "460D84E00000000000",
    "participant": "7000000012@c.us",
    "typeMessage": "pollMessage",
    "pollMessageData": {
      "name": "Выберите любимый цвет",
      "options": [],
      "multipleAnswers": true
    }
  },
  "statusMessage": "delivered",
  "sendByApi": false
}

```

```

{
  "type": "outgoing",
  "idMessage": "21D6C50B400DBCEC3000000000000000",
  "timestamp": 1706852882,
  "typeMessage": "quotedMessage",
  "chatId": "7000000012@c.us",
  "extendedTextMessage": {
    "text": "Кrypto",
    "stanzaId": "79DC2458992097F00000000000000000",
    "participant": "7000000012@c.us"
  },
  "quotedMessage": {
    "stanzaId": "79DC2458992097F00000000000000000",

```

```

"participant": "7000000012@c.us",
"typeMessage": "imageMessage", //"/videoMessage"/`documentMessage`/`audioMessage`/`stickerMessage`
"downloadUrl": "",
"caption": "",
"fileName": "",
"jpegThumbnail": "/9j/4AAQSkZJRgABAQAAQABAAD/2wCEAgq70/wD/2Q==",
"mimeType": "image/jpeg",
"isAnimated": false,
"isForwarded": false,
"forwardingScore": 0
},
"statusMessage": "read",
"sendByApi": true
}

```

Ошибки GetMessage

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
400	Bad Request Validation failed	Ошибка валидации
400	Validation failed. Details: 'idMessage' is not allowed to be empty	Ошибка валидации. Поле <code>idMessage</code> не должно быть пустым
400	Validation failed. Details: 'message id' must be at least 16 symbols	Ошибка валидации. Поле <code>idMessage</code> должно содержать минимум 16 символов
400	chatId not found	<code>chatID</code> не найдено. Вероятно, сообщение все еще находится в очереди сообщений, и система не успела его обработать. Вы можете запросить метод ShowMessagesQueue для проверки сообщений.
400	Message not found by id '{{idMessage}}'	<code>IDMessage</code> не найдено. Вероятно, сообщение все еще находится в очереди сообщений, и система не успела его обработать. Вы можете запросить метод ShowMessagesQueue для проверки сообщений. Вероятно, неверно указан чат, номер сообщения или инстанс на котором происходит запрос сообщения

LastIncomingMessages



Метод возвращает крайние входящие сообщения инстанса. По умолчанию возвращаются последние сообщения за 24 часа.

Чтобы получить **входящие сообщения**, включите настройку [Получать уведомления о входящих сообщениях и файлах](#) методом [SetSettings](#) или через [личный кабинет](#).

Применение настроек занимает до 5 минут.

Чтобы получить сообщения, отправленные и полученные с телефона **до подключения инстанса**, необходимо после сохранения настроек на инстансе сделать [разлогин](#) и [авторизоваться](#) повторно.

Метод LastIncomingMessages возвращает только ту историю, которую системе отдает WhatsApp.

Появления сообщений в журнале может занимать время до 2 минут, методы журналов необходимо использовать только для получения истории переписки. Для быстрого получения сообщений необходимо использовать [систему уведомлений](#).

ЗАПРОС

Для получения входящих сообщений требуется выполнить запрос по адресу:

```
GET {{apiUrl}}/waInstance{{idInstance}}/lastIncomingMessages/{{apiTokenInstance}}?minutes={{minutes_count}}
```

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры URL запроса

Параметр	Тип	Обязательный	Описание
<code>minutes</code>	integer	нет	Время в минутах, за которое требуется показать сообщения (по умолчанию равно 1440 минут)

ОТВЕТ

Поля ответа

Массив объектов с полями:

Поле	Тип	Описание
type	string	Вид сообщения, incoming - входящее сообщение
idMessage	string	Идентификатор входящего сообщения
timestamp	integer	Время принятия сообщения в UNIX-формате
typeMessage	string	Тип сообщения, возможные значения:
		textMessage - текстовое сообщение
		imageMessage - сообщение с изображением
		videoMessage - видео сообщение
		documentMessage - сообщение с файлом документа
		audioMessage - аудио сообщение
		stickerMessage - сообщение со стикером
		reactionMessage - сообщение с реакцией
		locationMessage - сообщение геолокации
		contactMessage - сообщение с контактом
		extendedTextMessage - сообщение со ссылкой и превью
		pollMessage - сообщение с опросом
		pollUpdateMessage - сообщение с обновлением опросом
		incomingBlock - уведомление о добавлении чата в список заблокированных контактов. (Уведомление временно не работает)
		quotedMessage - сообщение с цитированием
		buttonsMessage - сообщение с кнопками (Метод временно не работает)
		templateMessage - сообщение с шаблонными кнопками (Метод временно не работает)
		listMessage - сообщение с кнопкой со списком (Метод временно не работает)
		buttonsResponseMessage - ответ с кнопками (Метод временно не работает)
		templateButtonsReplyMessage - ответ с фигурными кнопками (Метод временно не работает)
		listResponseMessage - ответ со списком (Метод временно не работает)
chatId	string	Идентификатор чата, в котором получено сообщение. По умолчанию принимает значение 10000000000@c.us, если typeMessage = incomingBlock
senderId	string	Идентификатор отправителя сообщения. По умолчанию принимает значение 10000000000@c.us, если typeMessage = incomingBlock
senderName	string	Имя отправителя сообщения
senderContactName	string	Имя отправителя из списка контактов текущего аккаунта
isForwarded	boolean	Флаг указывает является ли сообщение пересланным
forwardingScore	integer	Количество пересылок сообщения
textMessage	string	Текст сообщения, если typeMessage = textMessage / extendedTextMessage

Поле	Тип	Описание
downloadUrl	string	Ссылка на скачивание файла, если typeMessage = imageMessage / videoMessage / documentMessage / audioMessage / stickerMessage
caption	string	Описание файла, если typeMessage = imageMessage / videoMessage / documentMessage / audioMessage / stickerMessage
fileName	string	Название файла, если typeMessage = imageMessage / videoMessage / documentMessage / audioMessage / stickerMessage . Поле заполняется корректно только для typeMessage = documentMessage , в остальных случаях название файла генерируется автоматически
jpegThumbnail	string	Превью изображения в base64 кодировке, если typeMessage = imageMessage / videoMessage / documentMessage / audioMessage / stickerMessage
mimeType	string	Тип файла, согласно классификации Media Types , если typeMessage = imageMessage / videoMessage / documentMessage / audioMessage / stickerMessage
isAnimated	boolean	Флаг указывает анимирован ли файл, если typeMessage = imageMessage / videoMessage / documentMessage / audioMessage / stickerMessage
chatState	string	Состояние чата, если typeMessage = incomingBlock
location	object	Объект о структуре локации, если typeMessage = locationMessage
contact	object	Объект о структуре контакта, если typeMessage = contactMessage
extendedTextMessage	object	Объект с текстовым сообщением (расширенный), если typeMessage = extendedTextMessage / quotedMessage
extendedTextMessageData	object	Объект с реакцией, если typeMessage = reactionMessage
pollMessageData	object	Объект о структуре данных опроса, если typeMessage = pollMessage / pollUpdateMessage
quotedMessage	object	Объект данных о цитируемом сообщении, если typeMessage = quotedMessage / reactionMessage

Поля объекта location :

Поле	Тип	Описание
nameLocation	string	Название локации
address	string	Адрес локации
latitude	double	Широта локации
longitude	double	Долгота локации
jpegThumbnail	string	Превью изображения в base64 кодировке
isForwarded	boolean	Флаг указывает является ли сообщение пересланным
forwardingScore	integer	Количество пересылок сообщения

Поля объекта `contact` :

Поле	Тип	Описание
<code>displayName</code>	string	Отображаемое имя контакта
<code>vcard</code>	string	Структура VCard (визитной карточки контакта)
<code>isForwarded</code>	boolean	Флаг указывает является ли сообщение пересланным
<code>forwardingScore</code>	integer	Количество пересылок сообщения

Поля объекта `extendedTextMessage` :

Поле	Тип	Описание
<code>text</code>	string	Текст ссылки
<code>description</code>	string	Описание ссылки
<code>title</code>	string	Заголовок ссылки
<code>previewType</code>	string	Тип превью ссылки. Значение <code>previewType: None</code> устарело
<code>jpegThumbnail</code>	string	Превью изображения в base64 кодировке
<code>forwardingScore</code>	integer	Количество пересылок сообщения
<code>isForwarded</code>	boolean	Флаг указывает является ли сообщение пересланным
<code>mediaType</code>	string	Тип медиа сообщения, например "IMAGE" (только для рекламных сообщений)
<code>sourceId</code>	string	Id рекламы (только для рекламных сообщений)
<code>sourceType</code>	string	Тип источника, например "ad" (только для рекламных сообщений)
<code>sourceUrl</code>	string	Ссылка на рекламу (только для рекламных сообщений)
<code>thumbnailUrl</code>	string	Ссылка на превью изображения (только для рекламных сообщений)
<code>stanzaId</code>	string	ID цитируемого сообщения. Присутствует только для <code>typeMessage = quotedMessage</code>
<code>participant</code>	string	ID чата получателя/отправителя. Присутствует только для <code>typeMessage = quotedMessage</code>

Поля объекта `extendedTextMessageData` :

Поле	Тип	Описание
<code>text</code>	string	Реакция (эмоджи) на сообщение

Поля объекта `pollMessageData` :

Поле	Тип	Описание
<code>stanzaId</code>	string	id сообщения с исходящим опросом. Присутствует только для <code>typeMessage = pollUpdateMessage</code>
<code>name</code>	string	Текст опроса
<code>options</code>	array	Массив данных о вариантах выбора. Присутствует только для <code>typeMessage = pollMessage</code>
<code>votes</code>	array	Массив данных ответов. Присутствует только для <code>typeMessage = pollUpdateMessage</code>
<code>multipleAnswers</code>	boolean	Флаг о разрешении нескольких ответов

Поля массива options

Параметр	Тип	Описание
optionName	string	Текст варианта выбора

Поля массива votes :

Параметр	Тип	Описание
optionName	string	Текст варианта выбора
optionVoters	array	Массив идентификаторов проголосовавших аккаунтов

Поля объекта quotedMessage :

Поле	Тип	Описание
stanzaId	string	ID цитируемого сообщения
participant	string	ID чата получателя
typeMessage	string	Тип цитируемого сообщения

Остальные поля объекта quotedMessage заполняются в зависимости от типа цитируемого сообщения и идентичны полям описанных выше

Пример тела ответа

```
[
  {
    "type": "incoming",
    "idMessage": "9DB14F14A253D33F4A9CD84123456789",
    "timestamp": 1706522263,
    "typeMessage": "textMessage",
    "chatId": "120363153000000000@g.us",
    "textMessage": "Привет",
    "senderId": "7000000012@c.us",
    "senderName": "Николай",
    "senderContactName": "Коля"
  },
  {
    "type": "incoming",
    "idMessage": "EAAE5A5CD000000000",
    "timestamp": 1706601620,
    "typeMessage": "imageMessage", //"/videoMessage"/documentMessage/audioMessage/stickerMessage
    "chatId": "120363153000000000@g.us",
    "downloadUrl": "{{APIUrl}}/waInstance1234/downloadFile/EA1BD1AE042DC4F3609867126309D67C.jpg",
    "caption": "Как тебе?",
    "fileName": "03963bac-6020-423c-80df-1aa123456789.jpg",
    "jpegThumbnail": "/9j/4AAQSkZJ/aAAgBAwEBPwCVYw/xAApEAACAx/2k0Vh/VKAoAH/LeFhX/+AAMA/9k=",
    "mimeType": "image/jpeg",
    "isAnimated": false,
    "isForwarded": true,
    "forwardingScore": 1,
    "senderId": "7000000012@c.us",
    "senderName": "Николай",
    "senderContactName": "Коля"
  },
  {
    "type": "incoming",
    "idMessage": "3A1EBA0F23004BD4158B",
    "timestamp": 1706601621,
    "typeMessage": "reactionMessage",
    "chatId": "7000000012@c.us",
    "extendedTextMessageData": {
      "text": "😄"
    },
    "quotedMessage": {
      "stanzaId": "B4D7D8A1A7D887D5580F000000000000",
      "participant": "7000000012@c.us",
    },
    "senderId": "7000000013@c.us",
    "senderName": "Николай",
    "senderContactName": "Коля"
  },
  {
    "type": "incoming",
    "idMessage": "3A8427EF85F000000000",
```

```

"timestamp": 1706674611,
"typeMessage": "LocationMessage",
"chatId": "70000000012@с.us",
"location": {
  "nameLocation": "Cafe",
  "address": "Астана",
  "jpegThumbnail": "/9j/4AAQSkZJRgABAQAAASABIAAD/4QBvYRX",
  "latitude": 51.12800000000000,
  "longitude": 71.43076200000000,
  "forwardingScore": 0,
  "isForwarded": false
},
"senderId": "70000000012@с.us",
"senderName": "Николай",
"senderContactName": "Коля"
},
{
  "type": "incoming",
  "idMessage": "3AA2C1D8356000000000",
  "timestamp": 1706675513,
  "typeMessage": "contactMessage",
  "chatId": "70000000012@с.us",
  "contact": {
    "displayName": "Виктор Петров",
    "vcard": "BEGIN:VCARD\r\nVERSION:3.0\r\nn:;Виктор Петров;;;\r\nFN:Виктор Петров\r\nTEL;type=CELL;type=VOICE;waid=70000000013:+7 000 000-00-13\r\nEND:VCARD",
    "forwardingScore": 0,
    "isForwarded": false
  },
  "senderId": "70000000012@с.us",
  "senderName": "Николай",
  "senderContactName": "Коля"
},
{
  "type": "incoming",
  "idMessage": "3A66D7F0B50000000000",
  "timestamp": 1706675928,
  "typeMessage": "extendedTextMessage",
  "chatId": "70000000012@с.us",
  "textMessage": "Hqhghqh\r\nhttps://green-api.com/",
  "extendedTextMessage": {
    "text": "Hqhghqh\r\nhttps://green-api.com/",
    "description": "GREEN API - интеграция с WhatsApp на любом языке программирования PHP, JavaScript, 1C, Python, Java, C#, VBA",
    "title": "Green API для отправки и получения сообщений Whatsapp",
    "previewType": "None",
    "jpegThumbnail": "/9j/4AAQSkZJRgABAQAAASAB9K=",
    "forwardingScore": 0,
    "isForwarded": false
  },
  "senderId": "70000000012@с.us",
  "senderName": "Николай",
  "senderContactName": "Коля"
},
{
  "type": "incoming",
  "idMessage": "3A71511C3A0000000000",
  "timestamp": 1706676278,
  "typeMessage": "pollMessage",
  "chatId": "70000000012@с.us",
  "pollMessageData": {
    "name": "Выберите любимый цвет",
    "options": [
      {
        "optionName": "Красный"
      },
      {
        "optionName": "Зеленый"
      }
    ],
    "multipleAnswers": true
  },
  "senderId": "70000000012@с.us",
  "senderName": "Николай",
  "senderContactName": "Коля"
},
{
  "type": "incoming",
  "idMessage": "3A0493ED900000000000",
  "timestamp": 1706676447,
  "typeMessage": "pollUpdateMessage",
  "chatId": "70000000012@с.us",
  "pollMessageData": {
    "name": "Куда пойдём?",
    "stanzaId": "46618B98B3B9DF50F123456789123456",
    "votes": [
      {
        "optionName": "Пляж",
        "optionVoters": [
          "70000000012@с.us"
        ]
      },
      {
        "optionName": "Горы",
        "optionVoters": []
      }
    ]
  }
}

```

```

    ],
    "multipleAnswers": true
  },
  "senderId": "7000000012@с.us",
  "senderName": "Николай",
  "senderContactName": "Коля"
},
{
  "type": "incoming",
  "typeMessage": "incomingBlock", //Уведомление временно не работает
  "timestamp": 1706676650,
  "chatId": "1000000000@с.us",
  "senderId": "1000000000@с.us",
  "chatState": "blocked"
},
{
  "type": "incoming",
  "idMessage": "3AB0E938BA000000000",
  "timestamp": 1706677013,
  "typeMessage": "quotedMessage",
  "chatId": "7000000012@с.us",
  "extendedTextMessage": {
    "text": "За пляж",
    "stanzaId": "46618B9883B9DF50F123456789123456",
    "participant": "7000000012@с.us"
  },
  "quotedMessage": {
    "stanzaId": "46618B9883B9DF50F123456789123456",
    "participant": "7000000012@с.us",
    "typeMessage": "pollMessage",
    "pollMessageData": {
      "name": "Куда пойдём?",
      "options": [],
      "multipleAnswers": true
    }
  },
  "senderId": "7000000012@с.us",
  "senderName": "Николай",
  "senderContactName": "Коля"
},
{
  "type": "incoming",
  "idMessage": "3A70B27A6020000000000",
  "timestamp": 1706677254,
  "typeMessage": "quotedMessage",
  "chatId": "7000000012@с.us",
  "extendedTextMessage": {
    "text": "Курто!",
    "stanzaId": "EAAE5A5CD0000000000",
    "participant": "7000000012@с.us"
  },
  "quotedMessage": {
    "stanzaId": "EAAE5A5CD0000000000",
    "participant": "7000000012@с.us",
    "typeMessage": "imageMessage", //"/videoMessage"/`documentMessage`/`audioMessage`/`stickerMessage`
    "downloadUrl": "",
    "caption": "",
    "fileName": "",
    "jpegThumbnail": "/9j/4AAQSkZJRgABAQAA=",
    "mimeType": "image/jpeg",
    "isAnimated": false,
    "isForwarded": false,
    "forwardingScore": 0
  },
  "senderId": "7000000012@с.us",
  "senderName": "Николай",
  "senderContactName": "Коля"
}
{
  "type": "incoming",
  "idMessage": "F7AEC1B7086ECD7E6E45923F5EDB825",
  "timestamp": 1728018697,
  "typeMessage": "extendedTextMessage",
  "chatId": "79001234568@с.us",
  "textMessage": "https://fb.me/gUdvBwGIH",
  "extendedTextMessage": {
    "text": "Новый инстанс для вашего бизнеса!",
    "description": "",
    "title": "Green API",
    "jpegThumbnail": "/9j/4AAQSkZJRgABAQAAQABAAD/7QCEUghvdG9zaG9wIDMuMAA4qIjUeUjMEjYwGc9BTFMXYVg5nm4ADy6HLLZDQn1Ye5RMev+ae1hED2B9dQ1ARwjdIwYcg4/r/9k=",
    "forwardingScore": 0,
    "isForwarded": false,
    "mediaType": "IMAGE",
    "sourceUrl": "https://fb.me/gUdvBwGIH",
    "thumbnailUrl": "https://scontent.xx.fbcdn.net/v/t39.30808-6/436945887_122115998378256688_307643393252141869_n.jpg?stp=dst-jpg_s851x315&nc_cat=111&ccb=1-7&nc_sid=5f2048&nc_ohc=x61r6W-VrJIAb69T4Dl&nc_ad=z-m&nc_cid=0&nc_ht=scontent.xx&oh=00_AfDlFkjNOBYEntBu6SyaU0xDuxbvarjKha9Ez5RQXjsfg&oe=662D7680"
  },
  "senderId": "79001234568@с.us",
  "senderName": "Николай",
  "senderContactName": "Коля"
}
}
]

```

Ошибки LastIncomingMessages

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

LastOutgoingMessages



Метод возвращает крайние отправленные сообщения аккаунта. По умолчанию возвращаются последние сообщения за 24 часа.

Чтобы получить **статусы отправленных сообщений**, включите настройку `Получать уведомления о статусах отправленных сообщений` методом `SetSettings` или через [личный кабинет](#).

Сообщения отправленные с API записываются в журнал автоматически.

Применение настроек занимает до 5 минут.

Чтобы получить сообщения, отправленные и полученные с телефона **до подключения инстанса**, необходимо после сохранения настроек на инстансе сделать [разлогин](#) и [авторизоваться](#) повторно.

Метод LastOutgoingMessages возвращает только ту историю, которую системе отдает WhatsApp.

Появления сообщений в журнале может занимать время до 2 минут, методы журналов необходимо использовать только для получения истории переписки. Для быстрого получения сообщений необходимо использовать систему уведомлений.

ЗАПРОС

Для получения отправленных сообщений требуется выполнить запрос по адресу:

```
GET {{apiUrl}}/waInstance{{idInstance}}/lastOutgoingMessages/{{apiTokenInstance}}
```

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры URL запроса

Параметр	Тип	Обязательный	Описание
<code>minutes</code>	integer	нет	время в минутах, за которое требуется показать сообщения (по умолчанию равно 1440 минут)

ОТВЕТ

Поля ответа

Массив объектов с полями:

Поле	Тип	Описание
type	string	Вид сообщения, <code>outgoing</code> - исходящее сообщение
idMessage	string	Идентификатор исходящего сообщения
timestamp	integer	Время крайнего действия по сообщению в UNIX-формате
statusMessage	string	Статус исходящего сообщения, возможные значения: <code>noAccount</code> - нет аккаунта WhatsApp на номере телефона <code>notInGroup</code> - не состоите в данной группе <code>pending</code> - сообщение отправляется <code>sent</code> - отправлено <code>delivered</code> - доставлено <code>read</code> - прочитано/просмотрено/прослушано <code>yellowCard</code> - приостановка отправки сообщений из-за спамерской активности. Сообщения отправленные после получения статуса хранятся в очереди к отправке 24 часа. Для продолжения работы инстанса требуется сделать перезагрузку инстанса
sendByApi	boolean	Флаг указывает отправлено сообщение из API или нет
typeMessage	string	Тип сообщения, возможные значения: <code>textMessage</code> - текстовое сообщение <code>imageMessage</code> - сообщение с изображением <code>videoMessage</code> - видео сообщение <code>documentMessage</code> - сообщение с файлом документа <code>audioMessage</code> - аудио сообщение <code>stickerMessage</code> - сообщение со стикером <code>reactionMessage</code> - сообщение с реакцией <code>locationMessage</code> - сообщение геолокации <code>contactMessage</code> - сообщение с контактом <code>extendedTextMessage</code> - сообщение со ссылкой и превью <code>pollMessage</code> - сообщение с опросом <code>quotedMessage</code> - сообщение с цитированием
chatId	string	Идентификатор чата , в который сообщение было отправлено
isForwarded	boolean	Флаг указывает является ли сообщение пересланным
forwardingScore	integer	Количество пересылок сообщения
textMessage	string	Текст сообщения, если <code>typeMessage = textMessage / extendedTextMessage</code>
downloadUrl	string	Ссылка на скачивание файла, если <code>typeMessage = imageMessage / videoMessage / documentMessage / audioMessage / stickerMessage</code>
caption	string	Описание файла, если <code>typeMessage = imageMessage / videoMessage / documentMessage / audioMessage / stickerMessage</code>

Поле	Тип	Описание
fileName	string	Название файла, если typeMessage = imageMessage / videoMessage / documentMessage / audioMessage / stickerMessage . Поле заполняется корректно только для typeMessage = documentMessage , в остальных случаях название файла генерируется автоматически
jpegThumbnail	string	Превью изображения в base64 кодировке, если typeMessage = imageMessage / videoMessage / documentMessage / audioMessage / stickerMessage
mimeType	string	Тип файла, согласно классификации Media Types , если typeMessage = imageMessage / videoMessage / documentMessage / audioMessage / stickerMessage
isAnimated	boolean	Флаг указывает анимирован ли файл, если typeMessage = imageMessage / videoMessage / documentMessage / audioMessage / stickerMessage
location	object	Объект о структуре локации, если typeMessage = locationMessage
contact	object	Объект о структуре контакта, если typeMessage = contactMessage
extendedTextMessage	object	Объект с текстовым сообщением (расширенный), если typeMessage = extendedTextMessage / quotedMessage
extendedTextMessageData	object	Объект с реакцией, если typeMessage = reactionMessage
pollMessageData	object	Объект о структуре данных опроса, если typeMessage = pollMessage
quotedMessage	object	Объект данных о цитируемом сообщении, если typeMessage = quotedMessage / reactionMessage

Поля объекта location :

Поле	Тип	Описание
nameLocation	string	Название локации
address	string	Адрес локации
latitude	double	Широта локации
longitude	double	Долгота локации
jpegThumbnail	string	Превью изображения в base64 кодировке
isForwarded	boolean	Флаг указывает является ли сообщение пересланным
forwardingScore	integer	Количество пересылок сообщения

Поля объекта contact :

Поле	Тип	Описание
displayName	string	Отображаемое имя контакта
vcard	string	Структура VCard (визитной карточки контакта)
isForwarded	boolean	Флаг указывает является ли сообщение пересланным
forwardingScore	integer	Количество пересылок сообщения

Поля объекта `extendedTextMessage` :

Поле	Тип	Описание
<code>text</code>	string	Текст ссылки
<code>description</code>	string	Описание ссылки
<code>title</code>	string	Заголовок ссылки
<code>previewType</code>	string	Тип превью ссылки. Значение <code>previewType: None</code> устарело
<code>jpegThumbnail</code>	string	Превью изображения в <code>base64</code> кодировке
<code>stanzaId</code>	string	ID цитируемого сообщения. Присутствует только для <code>typeMessage = quotedMessage</code>
<code>participant</code>	string	ID чата получателя/отправителя. Присутствует только для <code>typeMessage = quotedMessage</code>
<code>isForwarded</code>	boolean	Флаг указывает является ли сообщение пересланным
<code>forwardingScore</code>	integer	Количество пересылок сообщения

Поля объекта `extendedTextMessageData` :

Поле	Тип	Описание
<code>text</code>	string	Реакция (эмоджи) на сообщение

Поля объекта `pollMessageData` :

Поле	Тип	Описание
<code>name</code>	string	Текст опроса
<code>options</code>	array	Массив данных о вариантах выбора
<code>votes</code>	array	Массив данных ответов
<code>multipleAnswers</code>	boolean	Флаг о разрешении нескольких ответов

Поля массива `options`

Параметр	Тип	Описание
<code>optionName</code>	string	Текст варианта выбора

Поля массива `votes` :

Параметр	Тип	Описание
<code>optionName</code>	string	Текст варианта выбора
<code>optionVoters</code>	array	Массив идентификаторов проголосовавших аккаунтов

Поля объекта `quotedMessage` :

Поле	Тип	Описание
<code>stanzaId</code>	string	ID цитируемого сообщения
<code>participant</code>	string	ID чата получателя
<code>typeMessage</code>	string	Тип цитируемого сообщения

Остальные поля объекта `quotedMessage` заполняются в зависимости от типа цитируемого сообщения и идентичны полям описанных выше

Пример тела ответа

```
[
  {
    "type": "outgoing",
    "idMessage": "BAE5143000000000",
    "timestamp": 1706761225,
    "typeMessage": "textMessage",
    "chatId": "70000000012@с.us",
    "textMessage": "Привет",
    "statusMessage": "read",
    "sendByApi": true
  },
  {
    "type": "outgoing",
    "idMessage": "BAE5F94000000000",
    "timestamp": 1706765613,
    "typeMessage": "imageMessage", /*/"videoMessage"/"documentMessage"/"audioMessage"/"stickerMessage"*/
    "chatId": "70000000012@с.us",
    "downloadUrl": "https://api.green-api.com/waInstance1234/downloadFile/3EB0BDDC94BFD3D4FA.jpg",
    "caption": "Как тебе?",
    "fileName": "b89ded5e-1a3b-4168-895a-540123456789.jpg",
    "jpegThumbnail": "/9j/4AAQSkZJRgABAQAAQABAAQ/FFFCF//Z",
    "mimeType": "image/jpeg",
    "isAnimated": false,
    "isForwarded": true,
    "forwardingScore": 2,
    "statusMessage": "read",
    "sendByApi": false
  },
  {
    "type": "outgoing",
    "idMessage": "3A1EBA0F23004BD415BB",
    "timestamp": 1706601621,
    "typeMessage": "reactionMessage",
    "chatId": "70000000012@с.us",
    "extendedTextMessageData": {
      "text": "😄"
    },
    "quotedMessage": {
      "stanzaId": "B4D7D8A1A7D887D5580F000000000000",
      "participant": "70000000012@с.us",
    },
    "statusMessage": "read",
    "sendByApi": false
  },
  {
    "type": "outgoing",
    "idMessage": "BAE5870000000000",
    "timestamp": 1706761414,
    "typeMessage": "locationMessage",
    "chatId": "70000000012@с.us",
    "location": {
      "nameLocation": "Cafe",
      "address": "Астана",
      "jpegThumbnail": "/9j/4AAQSkZJRgABAQAAQABAAQ/4QBVRX",
      "latitude": 51.12800000000000,
      "longitude": 71.43076200000000,
      "forwardingScore": 0,
      "isForwarded": false
    },
    "statusMessage": "read",
    "sendByApi": true
  },
  {
    "type": "outgoing",
    "idMessage": "F39F2DCE0000000000",
    "timestamp": 1706762400,
    "typeMessage": "contactMessage",
    "chatId": "70000000012@с.us",
    "contact": {
      "displayName": "Виктор Петров",
      "vcard": "BEGIN:VCARD\nVERSION:3.0\nN:;Виктор Петров;;;\nFN:Виктор Петров\nTEL;type=CELL;type=VOICE;waid=70000000013:+7 000 000-00-13\nEND:VCARD",
      "forwardingScore": 0,
    }
  }
]
```

```

    "isForwarded": false
  },
  "statusMessage": "read",
  "sendByApi": false
},
{
  "type": "outgoing",
  "idMessage": "BAE56E0000000000",
  "timestamp": 1706849285,
  "typeMessage": "extendedTextMessage",
  "chatId": "7000000012@с.us",
  "textMessage": "https://green-api.com/",
  "extendedTextMessage": {
    "text": "https://green-api.com/",
    "description": "GREEN API - интеграция с WhatsApp на любом языке программирования PHP, JavaScript, 1C, Python, Java, C#, VBA",
    "title": "Green API для отправки и получения сообщений Whatsapp",
    "previewType": "None",
    "jpegThumbnail": "/9j/2wBDAAYEBQYFBAYGBQYHbWYICH/5IAMkIQnxOEIQkIEIQkIEIQkIEIQkIEIQkI//9k=",
    "forwardingScore": 0,
    "isForwarded": false
  },
  "statusMessage": "sent",
  "sendByApi": true
},
{
  "type": "outgoing",
  "idMessage": "460D84E00000000000",
  "timestamp": 1706762412,
  "typeMessage": "pollMessage",
  "chatId": "7000000012@с.us",
  "pollMessageData": {
    "name": "Выберите любимый цвет",
    "options": [
      {
        "optionName": "Красный"
      },
      {
        "optionName": "Зеленый"
      }
    ],
    "multipleAnswers": true
  },
  "statusMessage": "delivered",
  "sendByApi": true
},
{
  "type": "outgoing",
  "idMessage": "B1494991277FD000000000000000000",
  "timestamp": 1706849516,
  "typeMessage": "quotedMessage",
  "chatId": "7000000012@с.us",
  "extendedTextMessage": {
    "text": "Черный",
    "stanzaId": "460D84E00000000000",
    "participant": "7000000012@с.us"
  },
  "quotedMessage": {
    "stanzaId": "460D84E00000000000",
    "participant": "7000000012@с.us",
    "typeMessage": "pollMessage",
    "pollMessageData": {
      "name": "Выберите любимый цвет",
      "options": [],
      "multipleAnswers": true
    }
  },
  "statusMessage": "delivered",
  "sendByApi": false
},
{
  "type": "outgoing",
  "idMessage": "21D6C50B400DBCEC3000000000000",
  "timestamp": 1706852882,
  "typeMessage": "quotedMessage",
  "chatId": "7000000012@с.us",
  "extendedTextMessage": {
    "text": "Кrypto",
    "stanzaId": "79DC2458992097F00000000000000000",
    "participant": "7000000012@с.us"
  },
  "quotedMessage": {
    "stanzaId": "79DC2458992097F00000000000000000",
    "participant": "7000000012@с.us",
    "typeMessage": "imageMessage",
    "downloadUrl": "",
    "caption": "",
    "fileName": "",
    "jpegThumbnail": "/9j/4AAQSkZJRgABAQAAQABAAQ/2wCEAgg70/wD/2Q==",
    "mimeType": "image/jpeg",
    "isAnimated": false,
    "isForwarded": false,
    "forwardingScore": 0
  },
  "statusMessage": "read",

```

```
    "sendByApi": true  
  }  
]
```

Ошибки LastOutgoingMessages

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

1.3.6 Очереди

Очереди

В данном разделе представлены методы по работе с очередью сообщений к отправке

- [Получить очередь сообщений к отправке](#)
- [Очистить очередь сообщений к отправке](#)

ShowMessagesQueue

Метод предназначен для получения списка сообщений, находящихся в очереди на отправку. Сообщение на отправку хранится 24 часа в очереди и будет отправлено сразу же после авторизации телефона. Скорость отправки сообщений из очереди регулирует параметр [Интервал отправки сообщений](#).

ЗАПРОС

Для получения списка сообщений требуется выполнить запрос по адресу:

```
GET {{apiUrl}}/waInstance{{idInstance}}/showMessagesQueue/{{apiTokenInstance}}
```

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

ОТВЕТ**Поля ответа**

Массив объектов с полями:

Поле	Тип	Описание
messageID	string	ID сообщения, если <code>type = sendMessage / sendPoll / sendFileByUrl / sendLocation / sendContact / sendTextStatus / sendMediaStatus / sendVoiceStatus</code>
messagesIDs	array	Массив ID сообщений, если <code>type = ForwardMessages</code>
type	string	Тип запроса:
		<code>sendMessage</code> - текстовое сообщение
		<code>sendPoll</code> - сообщение с опросом
		<code>sendFileByUrl</code> - сообщение с файлом (сообщения отправленные методом <code>sendFileByUrl</code> , <code>sendFileByUpload</code>)
		<code>sendLocation</code> - сообщение с геопозицией
		<code>sendContact</code> - сообщение с контактом
		<code>sendLink</code> - сообщение с ссылкой (Не используется)
		<code>ForwardMessages</code> - сообщения к пересылке
		<code>sendVoiceStatus</code> - голосовой статус
		<code>sendMediaStatus</code> - статус с изображением или видео
		<code>sendTextStatus</code> - текстовой статус
body	object	Объект данных о сообщении в очереди

Поля объекта `body` :

Поле	Тип	Описание
<code>chatId</code>	string	Идентификатор чата в который сообщение будет отправлено
<code>message</code>	string	Текст сообщения, если <code>type = sendMessage / sendPoll / sendFileByUrl / sendLocation / sendContact / sendTextStatus</code>
<code>messages</code>	array	Массив пересылаемых сообщений, если <code>type = ForwardMessages</code>
<code>linkPreview</code>	boolean	Параметр включает отображение превью и описание ссылки, если <code>type = sendMessage / sendTextStatus</code>
<code>quotedMessageId</code>	string	Идентификатор цитируемого сообщения, если <code>type = sendMessage / sendPoll / sendFileByUrl / sendLocation / sendContact /</code>
<code>options</code>	array	Массив данных о вариантах выбора , если <code>type = sendPoll</code>
<code>fileName</code>	string	Имя отправляемого файла, если <code>type = sendFileByUrl / sendMediaStatus / sendVoiceStatus</code>
<code>caption</code>	string	Описание под файлом, если <code>type = sendFileByUrl / sendMediaStatus</code>
<code>urlFile</code>	string	Ссылка на отправляемый файл <code>type = sendFileByUrl / sendMediaStatus / sendVoiceStatus</code>
<code>archive</code>	string	Поле устарело, если <code>type = sendFileByUrl</code>
<code>latitude</code>	string	Широта локации, если <code>type = sendLocation</code>
<code>longitude</code>	string	Долгота локации, если <code>type = sendLocation</code>
<code>nameLocation</code>	string	Название локации, если <code>type = sendLocation</code>
<code>address</code>	string	Адрес локации, если <code>type = sendLocation</code>
<code>contact</code>	object	Объект данных о сообщении с контактом, если <code>type = contactMessage</code>
<code>backgroundColor</code>	string	Цвет фона сообщения, если <code>type = sendTextStatus / sendVoiceStatus</code>
<code>font</code>	string	Шрифт сообщения, если <code>type = sendTextStatus</code>
<code>participants</code>	array<string>	Массив строк с идентификаторами контактов, которым будет доступен статус, если <code>type = sendTextStatus / sendMediaStatus / sendVoiceStatus</code>
<code>urlLink</code>	string	URL отправляемой ссылки, если <code>type = sendLink</code> (Не используется)
<code>chatIdFrom</code>	string	Идентификатор чата с которого пересылается сообщение

Поля массива `options`

Параметр	Тип	Описание
<code>optionName</code>	string	Текст варианта выбора

Поля объекта `contact` :

Поле	Тип	Описание
<code>phoneContact</code>	string	Номер телефона контакта в международном формате (без +) 11 или 12 цифр
<code>firstName</code>	string	Имя контакта
<code>lastName</code>	string	Фамилия контакта
<code>middleName</code>	string	Отчество контакта
<code>company</code>	string	Название компании контакта

Пример тела ответа

```
[
  {
    "messageID": "BAE123456789",
    "type": "sendMessage",
    "body": {
      "chatId": "11001234567@ec.us",
      "message": "I use Green-API to send this message to you!",
      "linkPreview": true
    }
  },
  {
    "messageID": "BAE1234567123",
    "type": "sendMessage",
    "body": {
      "chatId": "11001234567@ec.us",
      "message": "Hello",
      "quotedMessageId": "BAE123456789",
      "linkPreview": true
    }
  },
  {
    "messageID": "BAE5DE8CA912345",
    "type": "sendPoll",
    "body": {
      "chatId": "11001234567@ec.us",
      "message": "Please choose the color:",
      "options": [
        {
          "optionName": "green"
        },
        {
          "optionName": "red"
        }
      ]
    }
  },
  {
    "messageID": "BAE1234567789",
    "type": "sendFileByUrl",
    "body": {
      "fileName": "test.jpg",
      "chatId": "11001234567@ec.us",
      "quotedMessageId": "BAE1234567159",
      "caption": "caption",
      "archive": false,
      "urlFile": "https://sw-media-out.storage.net/1101123456/5005fe15-23ee-43ef-8535-35e1dff8315c.jpg"
    }
  },
  {
    "messageID": "BAE1234561230",
    "type": "sendLocation",
    "body": {
      "chatId": "11001234567@ec.us",
      "latitude": 44.9370129,
      "longitude": 89.8728409,
      "nameLocation": "I'm here, come",
      "address": "613123, Perm"
    }
  },
  {
    "messageID": "BAE1234567456",
    "type": "sendContact",
    "body": {
      "chatId": "11001234567@ec.us",
      "contact": {
        "phoneContact": 79001234568,
        "firstName": "Артем",
        "lastName": "Евпаторийский",
        "middleName": "Петрович",
        "company": "Велосипед"
      }
    }
  },
  {
    "messagesIds": [
      "BAE1234561256",
      "BAE1234561278"
    ],
    "type": "ForwardMessages",
    "body": {
      "chatId": "11001234567@ec.us",
      "chatIdFrom": "12036304684212345@ec.us",
      "messages": [
        "BAE1234561296",
        "BAE1234561205"
      ]
    }
  },
  {
    "messageID": "BAE583864C6D427F",
    "type": "sendTextStatus",
  }
]
```

```

"body": {
  "message": "I use Green-API to send this Status!",
  "backgroundColor": "#228B22",
  "font": "SERIF",
  "participants": [ //статус доступен только указанным контактам
    "70000001234@с.us",
    "70000001235@с.us"
  ],
  "linkPreview": true
},
{
  "messageID": "BAE514E5B2E4B22D",
  "type": "sendVoiceStatus",
  "body": {
    "urlFile": "https://my.site.com/audio/music.mp3",
    "fileName": "music.mp3",
    "backgroundColor": "#228B22",
    "participants": [] // статус доступен всем контактам
  }
},
{
  "messageID": "BAE5D81B7C3EED1C",
  "type": "sendMediaStatus",
  "body": {
    "urlFile": "https://my.site.com/img/horse.png",
    "fileName": "horse.png",
    "caption": "Little horse",
    "participants": [] // статус доступен всем контактам
  }
}
]

```

Ошибки ShowMessagesQueue

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

ClearMessagesQueue



Метод предназначен для очистки очереди сообщений на отправку. Сообщение на отправку хранится 24 часа в очереди и будет отправлено сразу же после авторизации телефона. Скорость отправки сообщений из очереди регулирует параметр [Интервал отправки сообщений](#).

После блокировки номера и подключения нового требуется производить очистку очереди. Так как возможна отправка сообщений оставшихся в очереди, что может привести к блокировке нового номера.

ЗАПРОС

Для очистки очереди сообщений требуется выполнить запрос по адресу:

```
GET {{apiUrl}}/waInstance/{{idInstance}}/cClearMessagesQueue/{{apiTokenInstance}}
```

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

ОТВЕТ

Поля ответа

Поле	Тип	Описание
<code>isCleared</code>	boolean	Флаг очистки очереди сообщений

Пример тела ответа

```
{
  "isCleared": true
}
```

Ошибки ClearMessagesQueue

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

1.3.7 Группы

Группы

В данном разделе представлены методы по работе с групповыми чатами

Важно! Работа с групповыми чатами поддерживаются только в составе участников до 100 контактов в группе.

- [Создать группу](#)
- [Изменить имя группы](#)
- [Получить информацию о группе](#)
- [Добавить участника в группу](#)
- [Удалить участника из группы](#)
- [Назначить права администратора группы](#)
- [Отозвать права администратора группы](#)
- [Установить аватар группы](#)
- [Выйти из группы](#)

CreateGroup



Метод предназначен для создания группового чата. При создании группы требуется имитировать работу человека. Создавать группу требуется не чаще чем 1 группу каждые 5 минут.

ЗАПРОС

Для создания группового чата требуется выполнить запрос по адресу:

POST `{{apiUrl}}/waInstance{{idInstance}}/createGroup/{{apiTokenInstance}}`

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>groupName</code>	string	Да	Наименование нового группового чата. Максимальная длина поля 100 символов
<code>chatIds</code>	array	Да	Коллекция идентификаторов участников группы

Пример тела запроса

```
{
  "groupName": "Group created by Green API",
  "chatIds": [
    "79001234568@c.us",
    "79001234569@c.us"
  ]
}
```

ОТВЕТ

Поля ответа

Поле	Тип	Описание
<code>created</code>	boolean	Флаг создания группы
<code>chatId</code>	string	Идентификатор группового чата
<code>groupInviteLink</code>	string	Ссылка приглашения в группу

Пример тела ответа

```
{
  "created": true,
  "chatId": "120363043968066561@g.us",
  "groupInviteLink": "https://chat.whatsapp.com/xxxxxxxxxxxxxxxxxxxxx"
}
```

Ошибки CreateGroup

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
400	Bad Request Validation failed	Ошибка валидации
500		Ошибку возвращает WhatsApp, ограничивая пользователя в создании группы. Попробуйте создать группу с мобильного устройства, Вы получите предупреждение: "Не удастся создать группу, потому что вы очень быстро создали слишком много групп. Повторите попытку позже."
500	Internal Server Error Timed Out	В качестве одного или нескольких элементов <code>chatIds</code> указана группа

UpdateGroupName



Метод изменяет наименование группового чата.

ЗАПРОС

Для изменения наименования группового чата требуется выполнить запрос по адресу:

POST `{{apiUrl}}/waInstance{{idInstance}}/updateGroupName/{{apiTokenInstance}}`

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>groupId</code>	string	Да	Идентификатор группового чата
<code>groupName</code>	string	Да	Наименование группового чата. Максимальная длина поля 100 символов

Пример тела запроса

```
{
  "groupId": "120363043968066561@g.us",
  "groupName": "Group created by Green API+"
}
```

ОТВЕТ

Поля ответа

Поле	Тип	Описание
<code>updateGroupName</code>	boolean	Флаг изменения названия группы

Пример тела ответа

```
{
  "updateGroupName": true
}
```

Ошибки UpdateGroupName

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
200	<code>"updateGroupName": false</code>	Неверно указан <code>groupId</code>
400	<code>Bad Request</code> <code>Validation failed</code>	Ошибка валидации

GetGroupData

Метод получает данные группового чата.

ЗАПРОС

Для получения данных группового чата требуется выполнить запрос по адресу:

POST `{{apiUrl}}/waInstance{{idInstance}}/getGroupData/{{apiTokenInstance}}`

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>groupId</code>	string	Да	Идентификатор группового чата

"Примечание" Если пользователь не является администратором или владельцем группы, метод вернёт код 200 и пустое поле **groupInviteLink**. Для получения ссылки необходимо запрашивать метод с номера, являющегося владельцем группы, либо назначить текущий номер администратором.

Пример тела запроса

```
{
  "groupId": "120363043968066561@g.us"
}
```

ОТВЕТ**Поля ответа**

Поле	Тип	Описание
<code>groupId</code>	string	Идентификатор группового чата
<code>owner</code>	string	Идентификатор владельца группы
<code>subject</code>	string	Наименование группы
<code>creation</code>	integer	Время создания группы в Unix-формате
<code>participants</code>	array	Коллекция участников группы
<code>subjectTime</code>	integer	Время создания наименования группы в Unix-формате
<code>subjectOwner</code>	string	Идентификатор пользователя создавшего наименование группы
<code>groupInviteLink</code>	string	Ссылка приглашения в группу

Поля объектов из массива `participants`

Поле	Тип	Описание
<code>id</code>	string	Идентификатор участника группового чата
<code>isAdmin</code>	boolean	Флаг, является ли пользователь администратором группы
<code>isSuperAdmin</code>	boolean	Флаг, является ли пользователь супер администратором группы

Пример тела ответа

```
{
  "groupId": "120363043968066561@g.us",
  "owner": "11001234567@c.us",
  "subject": "Green API Group",
  "creation": 1587570015,
  "participants": [
    {
      "id": "11001234567@c.us",
      "isAdmin": true,
      "isSuperAdmin": true
    },
    {
      "id": "79001234568@c.us",
      "isAdmin": true,
      "isSuperAdmin": false
    },
    {
      "id": "79001234569@c.us",
      "isAdmin": false,
      "isSuperAdmin": false
    }
  ],
  "subjectTime": 1587737715,
  "subjectOwner": "11001234567@c.us",
  "groupInviteLink": "https://chat.whatsapp.com/xxxxxxxxxxxxxxxxxxxxxxx"
}
```

Ошибки GetGroupData

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
200	Error: forbidden	Вы не состоите в группе
200	Error: item-not-found	Группы не существует
400	Bad Request Validation failed	Ошибка валидации
400	Bad Request Validation failed. Details: 'groupId' must be the next formats: 'group_id@g.us'	Неверный формат поля groupId, поле указывается в формате group_id@g.us

AddGroupParticipant



Метод добавляет участника в групповой чат.

"Важно" Добавлять участников, менять название и вносить другие изменения может только администратор. Если вы не являетесь администратором группы, то метод вернёт ошибку.

ЗАПРОС

Для добавления участника в групповой чат требуется выполнить запрос по адресу:

```
POST {{apiUrL}}/waInstance{{idInstance}}/addGroupParticipant/{{apiTokenInstance}}
```

Для получения параметров запроса `apiUrL`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>groupId</code>	string	Да	Идентификатор группового чата
<code>participantChatId</code>	string	Да	Идентификатор участника, добавляемого в групповой чат.

Пример тела запроса

Добавление участника в групповой чат:

```
{
  "groupId": "120363043968066561@g.us",
  "participantChatId": "79001234565@с.us"
}
```

ОТВЕТ

Поля ответа

Поле	Тип	Описание
<code>addParticipant</code>	boolean	Флаг добавления участника в групповой чат

"Почему ответ запроса false?"

Ответ на запрос или значение `addParticipant` может быть `false` по трём причинам:

1. Пользователь, который добавляет нового участника в группу, не обладает статусом администратора данной группы.
2. У пользователя, который добавляет нового участника в группу, отсутствует сохраненный номер участника в телефонной книге.
3. Контакт уже состоит в группе.

Некоторые номера могут добавляться в группу и без сохранения в телефонную книгу.

Тем не менее, для обеспечения успешного добавления участника в группу рекомендуется всегда предварительно сохранять номер телефона участника в списке контактов.

Пример тела ответа

```
{
  "addParticipant": true
}
```

Ошибки AddGroupParticipant

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
200	"addParticipant": false	Указанный номер уже состоит в группе
200	"addParticipant": false	Вы не состоите в группе
200	"addParticipant": false	Некорректно указан participantChatId
400	Bad Request Validation failed	Ошибка валидации

RemoveGroupParticipant

Метод удаляет участника из группового чата.

ЗАПРОС

Для удаления участника из группового чата требуется выполнить запрос по адресу:

POST `{{apiUrl}}/waInstance{{idInstance}}/removeGroupParticipant/{{apiTokenInstance}}`

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>groupId</code>	string	Да	Идентификатор группового чата
<code>participantChatId</code>	string	Да	Идентификатор участника, удаляемого из группы

Пример тела запроса

Удаление участника группового чата:

```
{
  "groupId": "120363043968066561@g.us",
  "participantChatId": "79001234565@e.us"
}
```

ОТВЕТ**Поля ответа**

Поле	Тип	Описание
<code>removeParticipant</code>	boolean	Флаг удаления участника из группы

Пример тела ответа

```
{
  "removeParticipant": true
}
```

Ошибки RemoveGroupParticipant

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
200	<code>"removeParticipant": false</code>	Неверно указан <code>participantChatId</code>
200	<code>"removeParticipant": false</code>	Номер не состоит в группе
400	Bad Request Validation failed	Ошибка валидации

SetGroupAdmin



Метод назначает участника группового чата администратором.

ЗАПРОС

Для назначения участника группового чата администратором требуется выполнить запрос по адресу:

POST `{{apiUrl}}/waInstance{{idInstance}}/setGroupAdmin/{{apiTokenInstance}}`

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>groupId</code>	string	Да	Идентификатор группового чата
<code>participantChatId</code>	string	Да	Идентификатор участника группы, назначаемого в качестве администратора

Пример тела запроса

Назначение участника группового чата администратором:

```
{
  "groupId": "120363043968066561@.us",
  "participantChatId": "79001234565@.us"
}
```

ОТВЕТ

Поля ответа

Поле	Тип	Описание
<code>setGroupAdmin</code>	boolean	Флаг назначения участника группы администратором

Пример тела ответа

```
{
  "setGroupAdmin": true
}
```

Ошибки SetGroupAdmin

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
200	<code>failed to set group admin</code>	Неверно указан <code>groupId</code>
200	<code>participant not found</code>	Указанный номер не состоит в группе
400	<code>Bad Request</code> <code>Validation failed</code>	Ошибка валидации

RemoveAdmin

Метод лишает участника прав администрирования группового чата.

ЗАПРОС

Для лишения участника прав администрирования группового чата требуется выполнить запрос по адресу:

POST `{{apiUrl}}/waInstance{{idInstance}}/removeAdmin/{{apiTokenInstance}}`

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>groupId</code>	string	Да	Идентификатор группового чата
<code>participantChatId</code>	string	Да	Идентификатор участника группы , которого требуется лишить прав администрирования группы

Пример тела запроса

Лишение участника прав администрирования группы:

```
{
  "groupId": "120363043968066561@.us",
  "participantChatId": "79001234565@.us"
}
```

ОТВЕТ**Поля ответа**

Поле	Тип	Описание
<code>removeAdmin</code>	boolean	Флаг лишения участника группы прав администратора

Пример тела ответа

```
{
  "removeAdmin": true
}
```

Ошибки RemoveAdmin

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
200	<code>"removeAdmin": false</code>	Неверно указан <code>groupId</code>
200	<code>"removeAdmin": false</code>	Указанный номер не состоит в группе
400	Bad Request Validation failed	Ошибка валидации

SetGroupPicture



Метод устанавливает аватар группы.

ЗАПРОС

Для установки аватара группового чата требуется выполнить запрос по адресу:

```
POST {{apiUrl}}/waInstance{{idInstance}}/setGroupPicture/{{apiTokenInstance}}
```

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>file</code>	file	Да	Отправляемый файл в формате *.jpg
<code>groupId</code>	string	Да	Идентификатор группового чата

ОТВЕТ

Поля ответа

Поле	Тип	Описание
<code>setGroupPicture</code>	boolean	флаг результата установки аватара группы
<code>urlAvatar</code>	string	url на установленное изображение
<code>reason</code>	string	причина почему аватар не был установлен

Пример тела ответа

```
{
  "setGroupPicture": true,
  "urlAvatar": "https://pps.whatsapp.net/v/t61.24694-24/138639660_724754321806449_9118612187814397965_n.jpg?oh=997b0bb13b6bbb750432a86d4b8d935d&oe=600****B84",
  "reason": ""
}
```

Ошибки SetGroupPicture

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
200	{ "reason": "Timed Out", "urlAvatar": null, "setGroupPicture": false }	Некорректный тип данных поля <code>groupId</code> либо некорректный <code>groupId</code>
400	Bad Request Validation failed	Ошибка валидации
400	File is empty	Пустой файл
400	Unsupported media type	Неподдерживаемый формат файла

LeaveGroup



Метод производит выход пользователя текущего аккаунта из группового чата.

ЗАПРОС

Для выхода из группового чата требуется выполнить запрос по адресу:

POST `{{apiUrl}}/waInstance{{idInstance}}/LeaveGroup/{{apiTokenInstance}}`

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>groupId</code>	string	Да	Идентификатор группового чата, из которого необходимо выйти.

Пример тела запроса

```
{
  "groupId": "120363043968066561@eg.us"
}
```

ОТВЕТ

Поля ответа

Поле	Тип	Описание
<code>leaveGroup</code>	boolean	Флаг выхода из группы
<code>removeAdmin</code>	boolean	Флаг выхода из группы (если вы являетесь администратором группы)

Пример тела ответа

```
{
  "leaveGroup": true
}
```

```
{
  "removeAdmin": true
}
```

Ошибки LeaveGroup

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
400	Bad Request Validation failed	Ошибка валидации

1.3.8 Статусы (бета-версия)

Статусы (бета-версия)

"Бета-версия" Функциональность находится в режиме бета-версии. Функции могут быть изменены, а также могут работать нестабильно. В будущем за функционал может взиматься дополнительная плата.

Подключение к функционалу можно запросить через [техподдержку Green API](#)

Статусы продвигают товары и услуги через Вашу клиентскую базу WhatsApp. Также методы работы со статусами позволяют получать статистику отправки/доставки/просмотра статусов. Это позволяет оценивать эффективность Вашей клиентской базы и получать общее количество подписчиков WhatsApp, которые занесли Ваш номер себе в контакты. Рассылка сообщений только своим подписчикам позволяет значительно снизить риск блокировок.

СТАТУСЫ

- [Отправить текстовый статус](#)
- [Отправить голосовой статус](#)
- [Отправить медиа статус](#)
- [Удалить статус](#)

СТАТИСТИКА

- [Получить статистику статуса](#)

ИСТОРИЯ

- [Получить входящие статусы](#)
- [Получить исходящие статусы](#)

Статусы

SENDTEXTSTATUS



"Бета-версия" Функциональность находится в режиме бета-версии. Функции могут быть изменены, а также могут работать нестабильно. В будущем за функционал может взиматься дополнительная плата.

Подключение к функционалу можно запросить через [техподдержку Green API](#)

Метод предназначен для отправки текстового статуса. Статус будет добавлен в очередь на отправку. Статус на отправку хранится 24 часа в очереди и будет отправлен сразу же после авторизации телефона. Скорость отправки статусов из очереди регулирует параметр [Интервал отправки сообщений](#).

"Важно"

Чтобы получатель увидел статусы отправителя, обе стороны должны сохранить номера собеседников в список контактов

Список контактов подтягивается методом [GetContacts](#) по значению поля `contactName`. Чтобы получить актуальный список контактов, требуется переименовать контакт и заново авторизоваться, пересканировав QR код.

Статусы отправляются только первым 1024 контактам из метода [GetContacts](#) с корректным полем `contactName`.

Запрос

Для отправки текстового статуса требуется выполнить запрос по адресу:

POST `{{apiUrl}}/waInstance/{{idInstance}}/sendTextStatus/{{apiTokenInstance}}`

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
message	string	Да	Текст сообщения. Поддерживаются символы emoji 🤪. Требуется использовать кодировку UTF-8 без BOM
backgroundColor	string	Нет	Цвет фона сообщения. Значение по умолчанию: #FFFFFF. Пример сайта для получения значения цвета фона. Важно! Цвет фона необходимо поменять на любой, кроме белого. Текст статуса публикуется белыми символами.
font	string	Нет	Шрифт сообщения. Принимает значения: SERIF - Так будет выглядеть ваш текст SANS_SERIF - Так будет выглядеть ваш текст NORICAN_REGULAR - <i>Here is how your text will look</i> Шрифт применяется только для латинских букв BRYNDAN_WRITE - <i>Так будет выглядеть ваш текст</i> OSWALD_HEAVY - Так будет выглядеть ваш текст
participants	array<string>	Нет	Массив строк с идентификаторами контактов, которым будет доступен статус. Если значение поля пустое, "participants": [], статус будет доступен всем контактам

Максимальная длина текстового сообщения, поля message, составляет 500 символов

В случае добавления несуществующих номеров в поле participants статус не будет отправлен на эти номера

Пример тела запроса

Отправка текстового статуса:

```
{
  "message": "I use Green-API to send this Status!",
  "backgroundColor": "#228B22", // (#FFFFFF) фон белого цвета использовать не рекомендуется
  "font": "SERIF",
  "participants": ["70000001234@с.us", "440000001234@с.us"] // статус будет доступен только указанным контактам
}
```

Ответ

Поля ответа

Поле	Тип	Описание
idMessage	string	Идентификатор отправленного сообщения

Пример тела ответа

```
{
  "idMessage": "3EB0C767D097B7C7C030"
}
```

Ошибки SendTextStatus

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#).

Код HTTP	Идентификатор ошибки	Описание
403	Forbidden	Доступ к функционалу бета-версии методов статусов отсутствует. Подключение к функционалу можно запросить через техподдержку Green API

SENDVOICESTATUS



"Бета-версия" Функциональность находится в режиме бета-версии. Функции могут быть изменены, а также могут работать нестабильно. В будущем за функционал может взиматься дополнительная плата.

Подключение к функционалу можно запросить через [техподдержку Green API](#)

Метод предназначен для отправки голосового статуса.

Статус будет добавлен в очередь на отправку. Статус на отправку хранится 24 часа в очереди и будет отправлено сразу же после авторизации телефона.

Скорость отправки статусов из очереди регулирует параметр [Интервал отправки сообщений](#).

"Важно"

Чтобы получатель увидел статусы отправителя, обе стороны должны сохранить номера собеседников в список контактов

Список контактов подтягивается методом [GetContacts](#) по значению поля `contactName`. Чтобы получить актуальный список контактов, требуется переименовать контакт и заново авторизоваться, пересканировав QR код.

Статусы отправляются только первым 1024 контактам из метода [GetContacts](#) с корректным полем `contactName`.

Запрос

Для отправки голосового статуса требуется выполнить запрос по адресу:

POST `{{apiUrl}}/waInstance{{idInstance}}/sendVoiceStatus/{{apiTokenInstance}}`

"Ссылка должна быть прямой ссылкой на файл, а не ссылкой на скачивание файла" При использовании ссылки на скачивание файла вы получите код HTTP 200 OK, но статус не будет добавлен отправителю. Внешне ссылки могут быть идентичными, и их невозможно отличить без открытия ссылок в браузере.

****Как определить тип ссылки на файл?***

- * При переходе по прямой ссылке вы увидите свой файл в окне браузера, а при переходе по ссылке на скачивание файла начнется его загрузка.
- * При открытии инструментов разработчика легко заметить, что прямая ссылка на файл при его открытии обладает полем: `content-type: audio/wave` или идентичным, а ссылка для загрузки обладает полем: `content-type: application/octet-stream`.

****Как избежать проблем с внешними ссылками?***

- * Формировать ссылки при помощи метода `UploadFile`.

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>urlFile</code>	string	Да	Ссылка на отправляемый аудио файл
<code>fileName</code>	string	Да	Название файла. Должно содержать расширение файла. Требуется использовать кодировку UTF-8 без BOM. Поддерживаемые форматы: <code>.mp3</code> , <code>.wav</code>
<code>backgroundColor</code>	string	Нет	Цвет фона сообщения. Значение по умолчанию: <code>#FFFFFF</code> . Пример сайта для получения значения цвета фона
<code>participants</code>	array<string>	Нет	Массив строк с идентификаторами контактов, которым будет доступен статус. Если значение поля пустое, <code>"participants": []</code> , статус будет доступен всем контактам

В случае добавления несуществующих номеров в поле `participants` статус не будет отправлен на эти номера

Пример тела запроса

Отправка голосового статуса:

```
{
  "urlFile": "https://my.site.com/audio/music.mp3",
  "fileName": "music.mp3",
  "backgroundColor": "#228B22",
  "participants": ["70000001234@с.us", "440000001234@с.us"] // статус будет доступен только указанным контактам
}
```

Ответ

Поля ответа

Поле	Тип	Описание
idMessage	string	Идентификатор отправленного сообщения

Пример тела ответа

```
{
  "idMessage": "3EB0C767D097B7C7C030"
}
```

Ошибки SendVoiceStatus

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#).

Код HTTP	Идентификатор ошибки	Описание
403	Forbidden	Доступ к функционалу бета-версии методов статусов отсутствует. Подключение к функционалу можно запросить через техподдержку Green API

SENDMEDIASTATUS



"Бета-версия" Функциональность находится в режиме бета-версии. Функции могут быть изменены, а также могут работать нестабильно. В будущем за функционал может взиматься дополнительная плата.

Подключение к функционалу можно запросить через [техподдержку Green API](#)

Метод предназначен для отправки медиа-файлов. Статус будет добавлен в очередь на отправку. Статус на отправку хранится 24 часа в очереди и будет отправлено сразу же после авторизации телефона.

Скорость отправки статусов из очереди регулирует параметр [Интервал отправки сообщений](#).

"Важно"

Чтобы получатель увидел статусы отправителя, обе стороны должны сохранить номера собеседников в список контактов

Список контактов подтягивается методом [GetContacts](#) по значению поля `contactName`. Чтобы получить актуальный список контактов, требуется переименовать контакт и заново авторизоваться, пересканировав QR код.

Статусы отправляются только первым 1024 контактам из метода [GetContacts](#) с корректным полем `contactName`.

Для корректной работы метода рекомендуем использовать файлы со следующими параметрами:

- Рекомендуемый формат для аудио - MP3.
Если аудио имеет длину более одной минуты, при отправке оно будет обрезано до одной минуты.
- Рекомендуемый формат для видео - MP4.
Если видео имеет длину более одной минуты, при отправке оно будет обрезано до одной минуты.
- Рекомендуемое соотношение сторон для картинка 9:16 (вертикальная картинка)

Запрос

Для отправки медиа в статус требуется выполнить запрос по адресу:

POST `{{apiUrl}}/waInstance{{idInstance}}/sendMediaStatus/{{apiTokenInstance}}`

"Ссылка должна быть прямой ссылкой на файл, а не ссылкой на скачивание файла" При использовании ссылки на скачивание файла вы получите код HTTP 200 OK, но статус не будет добавлен отправителю. Внешне ссылки могут быть идентичными, и их невозможно отличить без открытия ссылок в браузере.

****Как определить тип ссылки на файл?***

- * При переходе по прямой ссылке вы увидите свой файл в окне браузера, а при переходе по ссылке на скачивание файла начнётся его загрузка.
- * При открытии инструментов разработчика легко заметить, что прямая ссылка на файл при его открытии обладает полем: ```content-type: image/jpeg``` или идентичным, а ссылка для загрузки обладает полем: ```content-type: application/octet-stream```.

****Как избежать проблем с внешними ссылками?***

- * Формировать ссылки при помощи метода `UploadFile`.

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
urlFile	string	Да	Ссылка на отправляемый файл
fileName	string	Да	Название файла. Должно содержать расширение файла. Требуется использовать кодировку UTF-8 без BOM. Например: test.jpg
caption	string	Нет	Описание медиа статуса. Максимальная длина поля 1024 символов.
participants	array<string>	Нет	Массив строк с идентификаторами контактов, которым будет доступен статус. Если значение поля пустое, "participants": [], статус будет доступен всем контактам

В случае добавления несуществующих номеров в поле participants статус не будет отправлен на эти номера

Пример тела запроса

Отправка медиа статуса:

```
{
  "urlFile": "https://my.site.com/img/horse.png",
  "fileName": "horse.png",
  "caption": "Little horse",
  "participants": ["70000001234@с.us", "440000001234@с.us"] // статус будет доступен только указанным контактам
}
```

Ответ

Поля ответа

Поле	Тип	Описание
idMessage	string	Идентификатор отправленного сообщения

Пример тела ответа

```
{
  "idMessage": "3EB0C767D097B7C7C030"
}
```

Ошибки SendMediaStatus

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#).

Код HTTP	Идентификатор ошибки	Описание
403	Forbidden	Доступ к функционалу бета-версии методов статусов отсутствует. Подключение к функционалу можно запросить через техподдержку Green API

DELETESTATUS



"Бета-версия" Функциональность находится в режиме бета-версии. Функции могут быть изменены, а также могут работать нестабильно. В будущем за функционал может взиматься дополнительная плата.

Подключение к функционалу можно запросить через [техподдержку Green API](#)

Метод предназначен для удаления статуса.

Запрос

Для удаления статуса требуется выполнить запрос по адресу:

POST `{{apiUrl}}/waInstance/{{idInstance}}/deleteStatus/{{apiTokenInstance}}`

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>idMessage</code>	string	Да	ID удаляемого сообщения

Пример тела запроса

```
{
  "idMessage": "BAE5F4886F6F2D05"
}
```

Ответ

Поля ответа

Тело ответа пустое. При успехе ответ сервера 200.

Ошибки DeleteStatus

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#).

Код HTTP	Идентификатор ошибки	Описание
400	<code>ID message notfound</code>	IDMessage не найдено
403	<code>Forbidden</code>	Доступ к функционалу бета-версии методов статусов отсутствует. Подключение к функционалу можно запросить через техподдержку Green API

Статистика

GETSTATUSSTATISTIC



"Бета-версия" Функциональность находится в режиме бета-версии. Функции могут быть изменены, а также могут работать нестабильно. В будущем за функционал может взиматься дополнительная плата.

Подключение к функционалу можно запросить через [техподдержку Green API](#)

Метод возвращает массив получателей со статусами, отмеченных как отправлено/доставлено/прочитано, для данного статуса.

Для получения статусов сообщений, необходимо включить настройку [Получать уведомления о статусах отправленных сообщений](#) методом [SetSettings](#) или через [личный кабинет](#) (статусы сообщения полученные до включения настройки не обновятся).

Применение настроек занимает до 5 минут.

Список контактов подтягивается методом [GetContacts](#) по значению поля `contactName`. Чтобы получить актуальный список контактов, требуется переименовать контакт и заново авторизоваться, пересканировав QR код.

Запрос

Для получения статистики статусов требуется выполнить запрос по адресу:

```
GET {{apiUrl}}/waInstance{{idInstance}}/getStatusStatistic/{{apiTokenInstance}}?idMessage={{id_Message}}
```

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры URL запроса

Параметр	Тип	Обязательный	Описание
<code>idMessage</code>	string	Да	ID сообщения статуса

Ответ

Поля ответа

Массив объектов с полями:

Поле	Тип	Описание
<code>participant</code>	array	Идентификатор контакта
<code>timestamp</code>	integer	Время крайнего действия по сообщению в UNIX-формате
<code>status</code>	string	Статус исходящего статуса, возможные значения:
		<code>pending</code> - сообщение отправляется
		<code>sent</code> - отправлено
		<code>delivered</code> - доставлено
		<code>read</code> - прочитано/просмотрено/прослушано

Пример тела ответа

```
[
  {
    "participant": "11001234567@e.us",
```

```

    "timestamp": 1587129319,
    "status": "read",
  },
  {
    "participant": "71234567891@c.us",
    "timestamp": 1587129319,
    "status": "delivered",
  },
]

```

Ошибки GetStatusStatistic

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#).

Код HTTP	Идентификатор ошибки	Описание
403	Forbidden	Доступ к функционалу бета-версии методов статусов отсутствует. Подключение к функционалу можно запросить через техподдержку Green API

История

GETINCOMINGSTATUSES



"Бета-версия" Функциональность находится в режиме бета-версии. Функции могут быть изменены, а также могут работать нестабильно. В будущем за функционал может взиматься дополнительная плата.

Подключение к функционалу можно запросить через [техподдержку Green API](#)

Метод возвращает крайние входящие статусы аккаунта. По умолчанию возвращаются статусы за последние 24 часа.

Статусы можно получить только от номеров из контактной книги.

Список контактов подтягивается методом [GetContacts](#) по значению поля `contactName`. Чтобы получить актуальный список контактов, требуется переименовать контакт и заново авторизоваться, пересканировав QR код.

Запрос

Для получения входящих статусов требуется выполнить запрос по адресу:

```
GET {{apiUrl}}/waInstance{{idInstance}}/getIncomingStatuses/{{apiTokenInstance}}?minutes={{minutes_count}}
```

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры URL запроса

Параметр	Тип	Обязательный	Описание
<code>minutes</code>	integer	Нет	время в минутах, за которое требуется показать статусы (по умолчанию равен 1440 минутам)

Ответ

Поля ответа

Массив объектов с полями:

Поле	Тип	Описание
type	string	Вид сообщения, <code>incoming</code> - входящее сообщение
idMessage	string	Идентификатор входящего сообщения
timestamp	integer	Время принятия сообщения в UNIX-формате
typeMessage	string	Тип сообщения, возможные значения: <code>extendedTextMessage</code> - Расширенное текстовое сообщение <code>imageMessage</code> - сообщение с изображением <code>videoMessage</code> - видео сообщение <code>audioMessage</code> - аудио сообщение
chatId	string	Идентификатор чата отправителя сообщения
senderId	string	Идентификатор отправителя сообщения
senderName	string	Имя отправителя сообщения
senderContactName	string	Имя отправителя из списка контактов текущего аккаунта
textMessage	string	Текст сообщения, если <code>typeMessage = extendedTextMessage</code>
downloadUrl	string	Ссылка на скачивание файла, если <code>typeMessage = imageMessage / videoMessage / audioMessage</code>
caption	string	Описание файла, если <code>typeMessage = imageMessage / videoMessage</code>
fileName	string	Название файла, если <code>typeMessage = imageMessage / videoMessage / audioMessage</code> . Поле генерируется автоматически
jpegThumbnail	string	Превью изображения в base64 кодировке, если <code>typeMessage = imageMessage / videoMessage</code>
mimeType	string	Тип файла, согласно классификации Media Types , если <code>typeMessage = imageMessage / videoMessage / audioMessage</code>
extendedTextMessage	object	Объект с текстовым сообщением (расширенный), если <code>typeMessage = extendedTextMessage</code>

Поля объекта `extendedTextMessage` :

Поле	Тип	Описание
<code>text</code>	string	Текст сообщения
<code>backgroundColor</code>	string	Цвет фона сообщения. Значение по умолчанию: #FFFFFF. Пример сайта для получения значения цвета фона
<code>font</code>	string	Шрифт сообщения. Принимает значения:
		<code>SERIF</code> - Так будет выглядеть ваш текст
		<code>SANS_SERIF</code> - Так будет выглядеть ваш текст
		<code>NORICAN_REGULAR</code> - <i>Here is how your text will look</i> Шрифт применяется только для латинских букв
		<code>BRYNDAN_WRITE</code> - <i>Так будет выглядеть ваш текст</i>
		<code>OSWALD_HEAVY</code> - Так будет выглядеть ваш текст

Пример тела ответа

```
[
  {
    "type": "incoming",
    "idMessage": "1E1A12D337F2BFA5FC000000000000",
    "timestamp": 1710232595,
    "typeMessage": "extendedTextMessage",
    "chatId": "7000000001@ec.us",
    "textMessage": "Прекрасное утро!",
    "extendedTextMessage": {
      "text": "Прекрасное утро!",
      "backgroundColor": "#228B22",
      "font": "SANS_SERIF"
    },
    "senderId": "7000000001@ec.us",
    "senderName": "Николай",
    "senderContactName": "Коля",
  },
  {
    "type": "incoming",
    "idMessage": "38E322274FDEBA56047000000000000",
    "timestamp": 1710232636,
    "typeMessage": "imageMessage", // "videoMessage" / "audioMessage"
    "chatId": "7000000001@ec.us",
    "downloadUrl": "https://sw-media.storage.greenapi.net/1101000000/DCF81410-bdbc-4aed-bf23-d1845c000000.jpg",
    "caption": "Прекрасное утро!",
    "fileName": "dcf81410-bdbc-4aed-bf23-d1845cd74754.jpg",
    "jpegThumbnail": "/9j/2wCEABALDA4MChAODQ4SERATGCGaGBYWGDEjJ",
    "mimeType": "image/jpeg",
    "senderId": "7000000001@ec.us",
    "senderName": "Николай",
    "senderContactName": "Коля",
  }
]
```

Ошибки `GetIncomingStatuses`

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#).

Код HTTP	Идентификатор ошибки	Описание
403	Forbidden	Доступ к функционалу бета-версии методов статусов отсутствует. Подключение к функционалу можно запросить через техподдержку Green API

GETOUTGOINGSTATUSES



"Бета-версия" Функциональность находится в режиме бета-версии. Функции могут быть изменены, а также могут работать нестабильно. В будущем за функционал может взиматься дополнительная плата.

Подключение к функционалу можно запросить через [техподдержку Green API](#)

Метод возвращает крайние отправленные статусы аккаунта. По умолчанию возвращаются статусы за последние 24 часа.

Для получения статусов сообщений, необходимо включить настройку [Получать уведомления о статусах отправленных сообщений](#) методом [SetSettings](#) или через [личный кабинет](#) (статусы сообщения полученные до включения настройки не обновятся).

Применение настроек занимает до 5 минут.

Статусы можно получить только от номеров из контактной книги.

Список контактов подтягивается методом [GetContacts](#) по значению поля `contactName`. Чтобы получить актуальный список контактов, требуется переименовать контакт и заново авторизоваться, пересканировав QR код.

Статусы отправляются только первым 1024 контактам из метода [GetContacts](#) с корректным полем `contactName`.

Запрос

Для получения статусов требуется выполнить запрос по адресу:

```
GET {{apiUrl}}/waInstance{{idInstance}}/getOutgoingStatuses/{{apiTokenInstance}}?minutes={{minutes_count}}
```

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры URL запроса

Параметр	Тип	Обязательный	Описание
<code>minutes</code>	integer	Нет	время в минутах, за которое требуется показать статусы (по умолчанию равен 1440 минутам)

Ответ

Поля ответа

Массив объектов с полями:

Поле	Тип	Описание
type	string	Вид сообщения, <code>outgoing</code> - исходящее сообщение
idMessage	string	Идентификатор исходящего сообщения
timestamp	integer	Время крайнего действия по сообщению в UNIX-формате
statusMessage	string	Статус исходящего сообщения, возможные значения: <code>pending</code> - сообщение отправляется <code>sent</code> - отправлено <code>delivered</code> - доставлено <code>read</code> - прочитано/просмотрено/прослушано
sendByApi	boolean	Флаг указывает отправлено сообщение из API или нет
typeMessage	string	Тип сообщения, возможные значения: <code>extendedTextMessage</code> - Расширенное текстовое сообщение <code>imageMessage</code> - сообщение с изображением <code>videoMessage</code> - видео сообщение <code>audioMessage</code> - аудио сообщение
chatId	string	Идентификатор чата отправителя сообщения
participants	array	Массив строк с идентификаторами контактов, которым будет доступен статус. Если значение поля пустое, "participants": [], статус доступен всем контактам. По умолчанию принимает значение [], если "sendByApi" = false
textMessage	string	Текст сообщения, если <code>typeMessage</code> = <code>extendedTextMessage</code>
downloadUrl	string	Ссылка на скачивание файла, если <code>typeMessage</code> = <code>imageMessage</code> / <code>videoMessage</code> / <code>audioMessage</code>
caption	string	Описание файла, если <code>typeMessage</code> = <code>imageMessage</code> / <code>videoMessage</code>
fileName	string	Название файла, если <code>typeMessage</code> = <code>imageMessage</code> / <code>videoMessage</code> / <code>audioMessage</code> . Поле генерируется автоматически
jpegThumbnail	string	Превью изображения в base64 кодировке, если <code>typeMessage</code> = <code>imageMessage</code> / <code>videoMessage</code>
mimeType	string	Тип файла, согласно классификации Media Types , если <code>typeMessage</code> = <code>imageMessage</code> / <code>videoMessage</code> / <code>audioMessage</code>
extendedTextMessage	object	Объект с текстовым сообщением (расширенный), если <code>typeMessage</code> = <code>extendedTextMessage</code>

Поля объекта `extendedTextMessage` :

Поле	Тип	Описание
<code>text</code>	string	Текст ссылки
<code>backgroundColor</code>	string	Цвет фона сообщения. Значение по умолчанию: #FFFFFF. Пример сайта для получения значения цвета фона
<code>font</code>	string	Шрифт сообщения. Принимает значения:
		<code>SERIF</code> - Так будет выглядеть ваш текст
		<code>SANS_SERIF</code> - Так будет выглядеть ваш текст
		<code>NORICAN_REGULAR</code> - <i>Here is how your text will look</i> Шрифт применяется только для латинских букв
		<code>BRYNDAN_WRITE</code> - <i>Так будет выглядеть ваш текст</i>
		<code>OSWALD_HEAVY</code> - Так будет выглядеть ваш текст
<code>participants</code>	array	Массив строк с идентификаторами контактов, которым будет доступен статус. Если значение поля пустое, <code>"participants": []</code> , статусы доступны всем контактам. По умолчанию принимает значение <code>[]</code> , если <code>"sendByApi" = false</code>

Пример тела ответа

```
[
  {
    "type": "outgoing",
    "idMessage": "BAE5018000000000",
    "timestamp": 1710134811,
    "typeMessage": "extendedTextMessage",
    "chatId": "70000000001@с.us",
    "textMessage": "I use Green-API to send this Status!",
    "extendedTextMessage": {
      "text": "I use Green-API to send this Status!",
      "backgroundColor": "#228B22",
      "font": "SERIF",
      "participants": [
        "70000000001@с.us",
        "70000000002@с.us" // статус будет доступен только указанным контактам
      ]
    },
    "statusMessage": "read",
    "sendByApi": true
  },
  {
    "type": "outgoing",
    "idMessage": "BAE5207000000000",
    "timestamp": 1710134814,
    "typeMessage": "imageMessage", // "videoMessage" / "audioMessage"
    "chatId": "70000000001@с.us",
    "participants": [], // статус доступен всем
    "downloadUrl": "https://sw-media.storage.greenapi.net/1101000000/DCF81410-bdbc-4aed-bf23-d1845c000000.jpg",
    "caption": "I use Green-API to send this Status!",
    "fileName": "DCF81410-bdbc-4aed-bf23-d1845cd74754.jpg",
    "jpegThumbnail": "/9j/2wCEABALDA4MChAODQ4SERATGcga6BYWGEjJ",
    "mimeType": "image/jpeg",
    "statusMessage": "read",
    "sendByApi": true
  }
]
```

Ошибки `GetOutgoingStatuses`

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#).

Код HTTP	Идентификатор ошибки	Описание
403	Forbidden	Доступ к функционалу бета-версии методов статусов отсутствует. Подключение к функционалу можно запросить через техподдержку Green API

1.3.9 Отметка прочтения

Отметка прочтения

В данном разделе представлены методы по работе с отметкой прочтения чатов

- [Отметить чат прочитанным](#)

ReadChat



Метод предназначен для отметки сообщений в выбранном чате прочитанными. Могут быть отмечены прочитанными все сообщения в чате или только одно выбранное сообщение.

Для корректной работы метода, требуется включить настройку инстанса [Получать уведомления о входящих сообщениях и файлах](#) через [личный кабинет](#) или методом [SetSettings](#)

Сообщения, полученные **после применения настроек**, получают статус **read**.

Сообщения, полученные **до применения настроек**, останутся в статусе **delivered**.

ЗАПРОС

Для отметки сообщений в чате прочитанными требуется выполнить запрос по адресу:

POST `{{apiUrl}}/waInstance{{idInstance}}/readChat/{{apiTokenInstance}}`

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>chatId</code>	string	Да	Идентификатор чата
<code>idMessage</code>	string	Нет	Идентификатор входящего сообщения, которое необходимо отметить прочитанным. Если не указан, то все непрочитанные сообщения в чате будут отмечены прочитанными.

Пример тела запроса

Отметка прочтения одного сообщения в чате:

```
{
  "chatId": "11001234567@c.us",
  "idMessage": "B275A7AA0D6EF89BB92451698DF174E6"
}
```

Отметка прочтения всех сообщений в чате:

```
{
  "chatId": "11001234567@c.us"
}
```

ОТВЕТ

Поля ответа

Поле	Тип	Описание
<code>setRead</code>	boolean	Флаг отметки сообщений прочитанными

Пример тела ответа

```
{
  "setRead": true
}
```

Ошибки ReadChatMessage

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
400	Bad Request Validation failed	Ошибка Валидации

1.3.10 Сервисные методы

Сервисные методы

В данном разделе представлены различные сервисные методы

- Проверить наличие WhatsApp
- Получить аватар
- Получить контакты
- Получить информацию о контакте
- Удалить сообщение
- Архивировать чат
- Разархивировать чат
- Изменить настройки исчезающих сообщений чата

CheckWhatsapp



Метод проверяет наличие аккаунта WhatsApp на номере телефона.

ЗАПРОС

Для проверки наличия аккаунта WhatsApp требуется выполнить запрос по адресу:

POST `{{apiUrl}}/waInstance{{idInstance}}/checkWhatsapp/{{apiTokenInstance}}`

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>phoneNumber</code>	integer	Да	Номер телефона получателя в международном формате: 11 или 12 цифр; Пример: 11001234567 или 380123456789

Пример тела запроса

```
{
  "phoneNumber": 11001234567
}
```

ОТВЕТ

Поля ответа

Поле	Тип	Описание
<code>existsWhatsapp</code>	boolean	Флаг наличия WhatsApp на номере телефона

Пример тела ответа

```
{
  "existsWhatsapp": true
}
```

Ошибки CheckWhatsapp

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
400	Bad Request Validation failed	Ошибка валидации
400	Validation failed. Details: 'value' must have at least 1 key	Поле <code>value</code> должно содержать минимум 1 символ
400	Validation failed. Details: Wrong format. 'phoneNumber' must contain only digits	Номер должен содержать только цифры
400	bad phone number, valid 11 or 12 digits	Неверный формат номера телефона, должен быть 11 или 12 цифр
400	check phone number timeout limit exceeded	Превышен лимит времени ожидания ответа о проверке номера телефона

GetAvatar

Метод возвращает аватар корреспондента или группового чата.

ЗАПРОС

Для получения аватара требуется выполнить запрос по адресу:

POST `{{apiUrl}}/waInstance{{idInstance}}/getAvatar/{{apiTokenInstance}}`

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>chatId</code>	string	Да	Идентификатор корреспондента или группового чата

Пример тела запроса

Для получения своего аватара - укажите в `chatId` свой номер ("`{ваш номер}@c.us`").

Получить аватар корреспондента:

```
{
  "chatId": "11001234567@c.us"
}
```

Получить аватар группового чата:

```
{
  "chatId": "120363043968066561@g.us"
}
```

ОТВЕТ**Поля ответа**

Поле	Тип	Описание
<code>urlAvatar</code>	string	Ссылка на аватар корреспондента или группового чата. Параметр принимает пустое значение в случае, если аватар не установлен
<code>available</code>	boolean	Флаг, который указывает на доступность аватара корреспондента или группового чата. Параметр принимает значение <code>false</code> , если у корреспондента включены приватность чата

Пример тела ответа

```
{
  "urlAvatar": "https://pps.whatsapp.net/v/link/to/the/image",
  "available": true
}
```

Ошибки GetAvatar

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
200	<pre>{ "urlAvatar": "", "available": false }</pre>	На номере установлены настройки конфиденциальности, ограничивающие доступ к аватару
200	<pre>{ "urlAvatar": "", "available": true }</pre>	Аватар не установлен либо на номере нет аккаунта WhatsApp
400	Bad Request Validation failed	Ошибка валидации

GetContacts

Метод предназначен для получения списка контактов текущего аккаунта.

Обновление информации о контактах может занимать до 5 минут.

При получении пустого массива данных - повторить вызов метода.

ЗАПРОС

Для получения списка контактов требуется выполнить запрос по адресу:

```
GET {{apiUrl}}/waInstance{{idInstance}}/getContacts/{{apiTokenInstance}}
```

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

ОТВЕТ**Поля ответа**

Поле	Тип	Описание
<code>id</code>	string	Идентификатор корреспондента или группового чата
<code>name</code>	string	Имя контакта. Возможные значения: 1) Если есть входящая переписка/реакции от аккаунта, то получаем имя из профиля WhatsApp 2) Если нет входящей переписки/реакций от аккаунта, то получаем пустую строку
<code>contactName</code>	string	Имя контакта из телефонной книги
<code>type</code>	string	Тип контакта. Возможные значения: <code>user</code> - контакт принадлежит корреспонденту <code>group</code> - контакт является групповым чатом

Пример тела ответа

```
[
  {
    "id": "11001234567@c.us",
    "name": "Иван Петров",
    "contactName": "Иван Петоров Работа",
    "type": "user"
  },
  {
    "id": "79001234568@c.us",
    "name": "Люся Сидорова",
    "contactName": "Люся Сидорова кружок шитья",
    "type": "user"
  },
  {
    "id": "79001234569-1479621234@g.us",
    "name": "Моя группа",
    "type": "group"
  }
]
```

Ошибки GetContacts

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Описание ошибки	Возможные решения
200	Метод getContacts возвращает пустой массив []	<ol style="list-style-type: none">1. Пересканировать QR-код.2. Обратиться в службу техподдержки.

GetContactInfo



Метод предназначен для получения информации о контакте.

ЗАПРОС

Для получения информации о контакте требуется выполнить запрос по адресу:

POST

`{{apiUrl}}/waInstance{{idInstance}}/getContactInfo/{{apiTokenInstance}}`

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>chatId</code>	string	Да	Идентификатор корреспондента

Пример тела запроса

```
{
  "chatId": "71234567890@c.us"
}
```

ОТВЕТ

Поля ответа

Поле	Тип	Описание
avatar	string	ссылка на аватар
name	string	Имя контакта. Возможные значения: 1) Если есть входящая переписка/реакции от аккаунта, то получаем имя из профиля WhatsApp; 2) Если нет входящей переписки/реакций от аккаунта, то получаем пустую строку.
contactName	string	Имя контакта из телефонной книги. Возможные значения: 1) Если номер сохранён в контактах, то получаем имя из контактной книги; 2) Если номер не добавлен или удалён из контактной книги, то получаем пустую строку.
email	string	Электронная почта контакта
category	string	Категория бизнес контакта
description	string	Описание бизнес контакта
products	object	Карточки товаров контакта
chatId	string	Идентификатор корреспондента
lastSeen	string	Время последнего статуса онлайн
isArchive	boolean	Статус архивации чата, принимает значения true/false
isDisappearing	boolean	Статус исчезающих сообщений чата, принимает значения true/false
isMute	boolean	Статус уведомлений чата, принимает значения true/false
messageExpiration	integer	Время жизни сообщений в чате, в секундах
muteExpiration	integer	Время, через сколько будут включены уведомления в чате
isBusiness	boolean	Флаг, который указывает использует ли контакт WhatsApp Business, принимает значения true/false

Поля объекта products

Параметр	Тип	Описание
id	string	id товара
imageUrls	object	Ссылки на изображения товара
availability	string	Доступность товара
reviewStatus	object	Статус валидации товара
name	string	Название товара
description	string	Описание товара
price	string	Цена товара
isHidden	boolean	Состояние товара

Поля объекта `imageUrls`

Параметр	Тип	Описание
<code>requested</code>	string	Ссылка на изображение товара с маленьким разрешением (превью)
<code>original</code>	string	Ссылка на изображение товара с большим разрешением (оригинал)

Поля объекта `reviewStatus`

Параметр	Тип	Описание
<code>whatsapp</code>	string	Товар одобрен WhatsApp-ом

Пример тела ответа

```
{
  "avatar": "https://pps.whatsapp.net/v/t61.24694-24/24_1349471992200940_2091838963901201896_n.jpg?ccb=11-4&oh=01_AVzZiLQn10nj9M9cfQV4PW5dgdX0ki0uD_jCqP2MCIpyA",
  "name": "Dealer",
  "contactName": "Dealer",
  "email": "24service@tt.tt",
  "category": "Automotive Dealership",
  "description": "Официальный сервис",
  "products": [
    {
      "id": "42079728159",
      "imageUrls": {
        "requested": "https://mmg.whatsapp.net/v/t45.5328-4/263329037_6625110154227932_2879714823340281709_n.jpg?stp=dst-jpg_p100x100&ccb=1-7&nc_sid=c48759&nc_ohc=NKICbZlqfPMAx9077mo&nc_ad=z-m&nc_cid=0&nc_ht=mmg.whatsapp.net&oh=01_AVwYzx7CckCFf8F8xIIZ5m2AGdeC8YTnLyd29",
        "original": "https://mmg.whatsapp.net/v/t45.5328-4/263329037_6625110154227932_2879714823340281709_n.jpg?ccb=1-7&nc_sid=c48759&nc_ohc=NKICbZlqfPMAx9077mo&nc_ad=z-m&nc_cid=0&nc_ht=mmg.whatsapp.net&oh=01_AVzn_09azpKNRs1iPiD0TqkGyk4D7HZFSQMeobvRiR"
      },
      "reviewStatus": {
        "whatsapp": "APPROVED"
      },
      "availability": "in stock",
      "name": "Замена",
      "description": "От 1000 р.",
      "price": null,
      "isHidden": false
    },
    {
      "id": "3545870328871389",
      "imageUrls": {
        "requested": "https://mmg.whatsapp.net/v/t45.5328-4/261250418_4513761695371199_1710541959703469822_n.jpg?stp=dst-jpg_p100x100&ccb=1-7&nc_sid=c48759&nc_ohc=eps8lAw2_3MAX_mW8K&nc_ad=z-m&nc_cid=0&nc_ht=mmg.whatsapp.net&oh=01_AVxT3Hnbr04qKZJS0eK4d8p-noZokqly9QbpYFK-c_8KSA&oe",
        "original": "https://mmg.whatsapp.net/v/t45.5328-4/261250418_4513761695371199_1710541959703469822_n.jpg?ccb=1-7&nc_sid=c48759&nc_ohc=eps8lAw2_3MAX_mW8K&nc_ad=z-m&nc_cid=0&nc_ht=mmg.whatsapp.net&oh=01_AVxZwTCmzof0BoZDmIUpD328CtpJmVEXGdVzew&o"
      },
      "reviewStatus": {
        "whatsapp": "APPROVED"
      },
      "availability": "in stock",
      "name": "Техническое обслуживание",
      "price": null,
      "isHidden": false
    }
  ],
  "chatId": "71234567890@e.us",
  "lastSeen": null,
  "isArchive": false,
  "isDisappearing": false,
  "isMute": false,
  "messageExpiration": 0,
  "muteExpiration": null,
  "isBusiness": true
}
```

Ошибки GetContactInfo

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
400	Bad Request Validation failed	Ошибка валидации
400	Bad Request1 Validation failed. Details: The method GetContactInfo does not support group chats, to work with groups, use the GetGroupData method	Пустое тело запроса либо неверное chatId либо метод был использован с groupId
500	Internal Server Error interface conversion: interface {} is bool, not string	Неверно указан тип данных поля chatId

deleteMessage

Метод удаляет сообщение из чата.

ЗАПРОС

Для удаления сообщения из чата требуется выполнить запрос по адресу:

POST `{{apiUrl}}/waInstance{{idInstance}}/deleteMessage/{{apiTokenInstance}}`

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>chatId</code>	string	Да	Идентификатор корреспондента или группового чата
<code>idMessage</code>	string	Да	ID удаляемого сообщения

Пример тела запроса

```
{
  "chatId": "120363043968066561@g.us",
  "idMessage": "BAE5F4886F6F2D05"
}
```

ОТВЕТ**Поля ответа**

Тело ответа пустое. При успехе ответ сервера 200.

Ошибки deleteMessage

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
400	<code>Bad Request</code> <code>Validation failed</code>	Ошибка валидации
400	<code>Message by id not found</code>	Отсутствует поле <code>message</code> либо сообщение уже было удалено
500	<code>Internal Server Error</code> <code>interface conversion: interface {} is bool, not string</code>	Неверно указан тип данных поля <code>chatId</code>

ArchiveChat

Метод архивирует чат. Архивировать можно чаты, в которых есть хотя бы одно входящее сообщение.

ЗАПРОС

Для архивации чата требуется выполнить запрос по адресу:

POST `{{apiUrl}}/waInstance{{idInstance}}/archiveChat/{{apiTokenInstance}}`

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

"Примечание" Для работы метода система должна видеть входящие сообщения. Для этого в [настройках инстанса](#) включите [Получать уведомления о входящих сообщениях и файлах](#) или воспользуйтесь методом [SetSettings](#)

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>chatId</code>	string	Да	Идентификатор корреспондента или группового чата

Пример тела запроса

```
{
  "chatId": "120363043968066561@eg.us"
}
```

ОТВЕТ

Поля ответа

Тело ответа пустое. При успехе ответ сервера 200.

Ошибки archiveChat

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
400	Bad Request Validation failed	Ошибка валидации
400	"ArchiveChatError: cannot archive chat cause last message not found in chat"	<code>chatID</code> не верен или в чате нет сообщений
400	Bad Request ArchiveChatError: chat with id 71234567890@.us already archive=true	Чат уже архивирован
400	Bad Request ArchiveChatError: chat with id 71234567890@.us not found	Указанного номера нет в контактах
400	Validation failed. Details: 'value' must have at least 1 key	Ошибка валидации. Поле <code>value</code> должно содержать минимум 1 символ
500	Internal Server Error interface conversion: interface {} is bool, not string	Неверно указан тип данных поля <code>chatId</code>

UnarchiveChat

Метод разархивирует чат.

ЗАПРОС

Для разархивации чата требуется выполнить запрос по адресу:

POST

```
{{apiUrl}}/waInstance{{idInstance}}/unarchiveChat/{{apiTokenInstance}}
```

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>chatId</code>	string	Да	Идентификатор корреспондента или группового чата

Пример тела запроса

```
{
  "chatId": "120363043968066561@g.us"
}
```

ОТВЕТ**Поля ответа**

Тело ответа пустое. При успехе ответ сервера 200.

Ошибки UnarchiveChat

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
400	Bad Request Validation failed	Ошибка валидации
400	"UnarchiveChatError: cannot archive chat cause last message not found in chat"	chatID не верен или в чате нет сообщений
400	Bad Request UnarchiveChatError: chat with id 71234567890@ec.us already archive=false	Чат уже разархивирован
400	Bad Request UnarchiveChatError: chat with id 71234567890@ec.us not found	Указанного номера нет в контактах
400	Validation failed. Details: 'value' must have at least 1 key	Ошибка валидации. Поле <code>value</code> должно содержать минимум 1 символ
500	Internal Server Error interface conversion: interface {} is bool, not string	Неверно указан тип данных поля <code>chatId</code>

SetDisappearingChat



Метод предназначен для изменения настроек исчезающих сообщений в чатах. Используются стандартные настройки приложения: 0 (выключено), 86400 (24 часа), 604800 (7 дней), 7776000 (90 дней).

ЗАПРОС

Для установки настроек требуется выполнить запрос по адресу:

POST `{{apiUrl}}/waInstance{{idInstance}}/setDisappearingChat/{{apiTokenInstance}}`

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

Параметры запроса

Параметр	Тип	Обязательный	Описание
<code>chatId</code>	string	Да	Идентификатор корреспондента
<code>ephemeralExpiration</code>	integer	Да	Время жизни сообщений в чате, принимает значения в секундах: 0, 86400, 604800, 7776000

Пример тела запроса

```
{
  "chatId": "71234567890@c.us",
  "ephemeralExpiration": 0
}
```

ОТВЕТ

Поля ответа

Поле	Тип	Описание
<code>chatId</code>	string	Да
<code>disappearingMessagesInChat</code>	boolean	Состояние чата (исчезающий или обычный) принимает значения: true, false
<code>ephemeralExpiration</code>	integer	Время жизни сообщений в чате, принимает значения в секундах: 0, 86400, 604800, 7776000

Пример тела ответа

```
{
  "chatId": "712345678910@c.us",
  "disappearingMessagesInChat": false,
  "ephemeralExpiration": 0
}
```

Ошибки SetDisappearingChat

Перечень общих для всех методов ошибок смотрите в разделе [Стандартные ошибки](#)

Код HTTP	Идентификатор ошибки	Описание
400	Bad Request Validation failed	Ошибка валидации
400	Validation failed. Details: 'value' must have at least 2 keys	Ошибка валидации. Поле 'value' должно содержать минимум 2 символа
400	Internal Server Error Validation failed. Details: 'ephemeralExpiration' must be one of [0, 86400, 604800, 7776000]	Некорректные данные поля <code>ephemeralExpiration</code> , допустимые значения: 0, 86400, 604800, 7776000
500	Internal Server Error interface conversion: interface {} is bool, not string	Неверно указан тип данных поля <code>chatId</code>

1.3.11 Прочее

Идентификатор чата и корреспондента

В Green API поддерживаются два вида чатов - [личный чат](#) и [групповой чат](#). Личный чат используется для отправки персональных сообщений получателю. Групповой чат используется для организации общения нескольких участников в одной группе. Для работы с групповыми чатами в Green API предусмотрены соответствующие методы. Перед отправкой сообщения в личный или групповой чат требуется получить идентификатор чата.

ИДЕНТИФИКАТОР КОРРЕСПОНДЕНТА

Идентификатор корреспондента используется для идентификации получателя в исходящих сообщениях и для идентификации отправителя во входящих сообщениях, а также в любых других методах API, в которых требуется идентифицировать корреспондента. Формат идентификатора корреспондента совпадает с идентификатором [личного чата](#).

ЛИЧНЫЙ ЧАТ

Формат идентификатора личного чата формируется по шаблону `0000000000@с.us`, где вместо нулей `0000000000` требуется использовать номер телефона получателя. Телефон следует указывать полностью, с кодом страны и без пробелов. Например:

```
11001234567@с.us
```

ГРУППОВОЙ ЧАТ

Идентификатор группового чата представляет собой строку вида `XXXXXXXX-XXXXXXXX@г.us` или `XXXXXXXXXXXXXXXX@г.us`. Например:

```
11001234567-1581234040@г.us  
120363043968066561@г.us
```

Идентификатор группового чата может быть получен различными методами Green API, например:

- [Создать группу](#)
- [Получить журнал отправленных сообщений](#)
- [Получить журнал входящих сообщений](#)
- [Входящее сообщение: Текст](#)
- и др.

Важно

Идентификатор группового чата **НЕ требуется** формировать самостоятельно. Он формируется системой автоматически и возвращается различными методами Green API.

Интервал отправки сообщений

При отправке сообщений или файлов все данные помещаются в очередь на отправку. Все сообщения отправляются из очереди последовательно в порядке поступления в очередь **FIFO**. При этом устанавливается интервал между отправкой сообщений из очереди. Для изменения интервала отправки сообщений предназначен метод [SetSettings](#) и параметр `delaySendMessagesMilliseconds`. Отсчёт интервала происходит с момента отправки предыдущего сообщения из очереди. Соответственно, если прошло времени больше, чем значение интервала, то сообщение будет отправлено без задержки.

Минимальный интервал отправки сообщений составляет 500 мсек.

При выставлении интервала более 86400000 мсек сообщение не будет отправлено. Срок хранения сообщения в очереди не более 24 часов.

Посмотреть текущий интервал отправки можно методом [GetSettings](#), параметр `delaySendMessagesMilliseconds`.

ИЗМЕНЕНИЕ ИНТЕРВАЛА ОТПРАВКИ СООБЩЕНИЙ

Для изменения интервала отправки сообщений требуется выполнить запрос по адресу:

```
POST {{apiUrl}}/waInstance{{idInstance}}/setSettings/{{apiTokenInstance}}
```

Для получения параметров запроса `apiUrl`, `idInstance` и `apiTokenInstance` обратитесь к разделу [Перед началом работы](#).

В теле запроса требуется указать только один параметр `delaySendMessagesMilliseconds`.

Пример тела запроса для установки интервала отправки 5 сек

```
{
  "delaySendMessagesMilliseconds": 5000
}
```

ПРИМЕРЫ КОДА

Python **cURL** **Java - Spring** **Java - Unirest**

```
import requests

url = "{{apiUrl}}/waInstance{{idInstance}}/SetSettings/{{apiTokenInstance}}"

payload = "{\r\n\t\"delaySendMessagesMilliseconds\": 5000\r\n}\r\n"
headers = {
    'Content-Type': 'application/json'
}

response = requests.post(url, json=payload)

print(response.text.encode('utf8'))

curl --location '{{apiUrl}}/waInstance{{idInstance}}/setSettings/{{apiTokenInstance}}' \
--header 'Content-Type: application/json' \
--data '{
"delaySendMessagesMilliseconds": 5000
}'

var restTemplate = new RestTemplate();
var requestUrl = new StringBuilder();
requestUrl
    .append('{{apiUrl}}')
    .append("/waInstance").append('{{idInstance}}')
    .append("/setSettings/")
    .append('{{apiTokenInstance}}');

var headers = new HttpHeaders();
headers.setContentType(MediaType.APPLICATION_JSON);

var jsonBody = "{\r\n\"delaySendMessagesMilliseconds\": 15000}";

var requestEntity = new HttpEntity<>(jsonBody, headers);

var response = restTemplate.exchange(requestUrl.toString(), HttpMethod.POST, requestEntity, String.class);
System.out.println(response);

var requestUrl = new StringBuilder();
requestUrl
    .append('{{apiUrl}}')
    .append("/waInstance").append('{{idInstance}}')
    .append("/setSettings/")
    .append('{{apiTokenInstance}}');

var response = Unirest.post(requestUrl.toString())
    .header("Content-Type", "application/json")
    .body("{\r\n\"delaySendMessagesMilliseconds\": 15000}")
    .asString();

System.out.println(response);
```


Стандартные ошибки

НТТР ОШИБКИ

Код НТТР	Идентификатор ошибки	Описание
204	No Content	Использован метод запроса <code>OPTIONS</code> , измените метод запроса на <code>GET</code>
400	instance in starting process try later	Инстанс находится в процессе запуска/перезапуска. Попробуйте повторить попытку спустя несколько секунд
400	instance is starting or not authorized	Инстанс не авторизован. Для авторизации инстанса перейдите в личный кабинет и считайте QR-код из приложения WhatsApp Business на телефоне
400	Instance account is expired. Renew your instance from personal area	Срок действия инстанса истек. Продлите свой инстанс из личного кабинета
400	bad request data	Данные запроса не валидны. Исправьте ошибку в параметрах запроса и повторите попытку
400	Instance is deleted	Инстанс был удалён
400	Bad Request Validation failed	Ошибка валидации, подробнее в таблице
400	Unexpected token _ in JSON at position XX	Ошибка в структуре JSON
401	Unauthorized	Проблема с авторизацией, проверьте корректность указания <code>apiTokenInstance</code> , <code>partnerToken</code>
403	Forbidden	Проблема с аутентификацией, проверьте корректность указания <code>idInstance</code> и адрес запроса
404	Not Found	Некорректный метод запроса
429	Too Many Requests	Пользователь отправил слишком много запросов за заданный промежуток времени. Уменьшите частоту запросов. Рекомендации по частоте запросов
466	correspondentsStatus	Исчерпан лимит, подробнее в теле ошибки. Пример тела ошибки
499	Client Closed Request	Пользователь закрыл соединение, пока сервер обрабатывал запрос. Требуется увеличить время ожидания ответа от сервера и повторить запрос с задержкой. Если ошибка повторится, то информировать оператора и дать возможность повторить отправку
500	File from url exceeded max upload size. Size: XXXXmb Limit: 100mb Url:	Попытка отправки файла размером более 100 МБайт
500	request entity too large	Превышение допустимой длины json (>100кб)
502	Bad Gateway	Сервер не способен получить ответ от целевого сервера. Требуется 3 раза повторить запрос с задержкой. Если ошибка повторится, то информировать оператора и дать возможность повторить отправку

ОШИБКИ В ТЕЛЕ УВЕДОМЛЕНИЙ И ЖУРНАЛАХ

Код Ошибки	Поле	Описание	Решение
{{SWE001}}	textMessage	Ошибка получения первого входящего сообщения в чате при отправке на дополнительные устройства со стороны WhatsApp, наблюдается так же в нативных веб и десктоп версии. Фактически на все дополнительные устройства сообщение приходит пустое. Пример тела уведомления	Отлавливать текст-маркер {{SWE001}} в теле сообщения и настроить бота на повторный запрос сообщения у отправителя с текстом: "Пожалуйста, отправьте сообщение еще раз, я не смогла увидеть ваш ответ"
{{SWE002}}	downloadUrl	Ошибка получения большого файла более 100 МБайт. Пример тела уведомления	Отлавливать текст-маркер {{SWE002}} в теле сообщения и проверить сообщение в телефоне
{{SWE003}}	textMessage	Ошибка расшифровки сообщений, вызванная потерей актуальности ключей авторизации. Пример тела уведомления	Отлавливать текст-маркер {{SWE003}}. При появлении таких сообщений необходимо удалить все авторизации в телефоне и заново считать QR - код
{{SWE004}}	textMessage	Ошибка работы с групповыми чатами, вызванная превышением состава участников 100 контактов в группе. Пример тела уведомления	Отлавливать текст-маркер {{SWE004}}. При появлении таких сообщений необходимо удалить группы, в которых количество участников превышает 100 контактов или удалить контакты из групп
{{SWE999}}	textMessage	Ошибка получения входящего сообщения без ключей шифрования в чате со стороны WhatsApp. Фактически на все дополнительные устройства сообщение приходит пустое. Пример тела уведомления	Отлавливать текст-маркер {{SWE999}} в теле сообщения и настроить бота на повторный запрос сообщения у отправителя с текстом: "Пожалуйста, отправьте сообщение еще раз, я не смогла увидеть ваш ответ"

ОШИБКИ ВАЛИДАЦИИ (400: VALIDATION FAILED)

Идентификатор ошибки	Описание
'chatId' must be one of the next formats: 'phone_number@c.us' or 'group_id@g.us	Неверный формат поля <code>chatId</code> , поле указывается в формате <code>phone_number@c.us</code> либо <code>group_id@g.us</code>
'wrongFieldName' is not allowed	Недопустимое поле в теле запроса
'{{field}}' is not allowed	Недопустимое поле в теле запроса
'{{field}}' is required	Отсутствует обязательное поле в теле запроса
'{{field}}' is not allowed to be empty	Обязательное поле не может быть пустым
'message id' must be at least 16 symbols	id цитируемого сообщения должно быть не меньше 16 символов, проверьте правильность указанного <code>idmessage</code> или <code>quotedMessageId</code>
'XXX' must be a string	Неверный тип данных поля <code>XXX</code> - поле должно быть типа <code>string</code> . Поля данного типа: <code>message</code> , <code>idMessage</code> , <code>quotedMessageId</code> , <code>groupId</code> , <code>groupName</code> , <code>participantChatId</code> , <code>options[X].optionName</code> , <code>urlFile</code>
'XXX' must be a boolean	Неверный тип данных поля <code>XXX</code> - поле должно быть типа <code>boolean</code> . Поля данного типа: <code>LinkPreview</code> , <code>multipleAnswers</code>
'XXX' must be an array	Неверный тип данных поля <code>XXX</code> - поле должно быть типа <code>array</code> . Поля данного типа: <code>chatIds</code> , <code>options</code> , <code>messages</code>

ОШИБКИ API



[Список ошибок API и способы их обхода](#)

Достижение лимитов на тарифе Разработчик

В данном разделе предоставлены примеры тел ошибки 466 при достижении лимитов на тарифе Разработчик

ФОРМАТ ОШИБКИ 466

Параметр	Тип	Описание
invokeStatus	object	Объект данных о статусе лимита
correspondentsStatus	object	Объект данных о статусе лимита чатов

Поля объекта invokeStatus

Параметр	Тип	Описание
method	string	Название метода или лимита
used	string	Количество запросов метода за месяц
total	string	Лимит запросов метода за месяц. Подробнее об ограничениях каждого метода
status	string	Статус лимита
description	string	Описание лимита, если status = QUOTE_EXCEEDED

Поля объекта correspondentsStatus

Параметр	Тип	Описание
method	string	Название лимита. По умолчанию принимает значение correspondents
used	string	Количество использованных чатов за месяц
total	string	Общее количество чатов, которых можно использовать в месяц. По умолчанию принимает значение 3
status	string	Статус лимита чатов
description	string	Описание лимита чатов, если status = QUOTE_EXCEEDED

ПРИМЕР ТЕЛА ОШИБКИ 466 ПО МЕТОДАМ API

```
{
  "invokeStatus": {
    "method": "checkwhatsapp",
    "used": 100,
    "total": 100,
    "status": "QUOTE_EXCEEDED",
    "description": "Monthly quota has been exceeded. Please go to your console and change the tariff to business https://console.green-api.com"
  }
}
```

ПРИМЕР ТЕЛА ОШИБКИ 466 ПО КОЛИЧЕСТВУ ЧАТОВ

```
{
  "invokeStatus": {
    "method": "sendmessage",
    "used": 67,
    "total": 0,
    "status": "QUOTE_ALLOWED"
  },
  "correspondentsStatus": {
    "method": "correspondents",
    "used": 3,
    "total": 3,
    "status": "CORRESPONDENTS_QUOTE_EXCEEDED",
    "description": "Monthly quota has been exceeded. You can only send or receive messages from these numbers: 12345678901@c.us,12345678910@c.us,12345678911@c.us. Please go to your personal account and change the tariff to business https://console.green-api.com"
  }
}
```

1.3.12 Ограничение частоты запросов

Система предназначена для предотвращения неверного использования API. Запросы выполненные с частотой выше, чем указано в таблице будут отклонены с ошибкой 429. Значения применяются к каждому инстансу.

"Пример"

Для 10 инстансов, каждый из которых выполняет по 50 запросов SendMessage, вы можете суммарно выполнить 500 запросов без получения ошибки 429.

Если на одном инстансе будет выполнен 51 запрос SendMessage за 1 секунду, то 51-й запрос будет отклонен с ошибкой 429. В это же время остальные инстансы продолжат выполнять запросы без получения ошибок.

Допустимое число запросов

Метод	Число запросов в секунду
getSettings	1
setSettings	1
getStateInstance	1
getStatusInstance	1
reboot	1
logout	1
qr	3
scanQrCode	1
setProfilePicture	0,1
getWaSettings	1
sendFileByUpload	50
receiveNotification	100
deleteNotification	100
downloadFile	5
getChatHistory	1
getMessage	10
lastIncomingMessages	1
lastOutgoingMessages	1
createGroup	1
updateGroupName	1
getGroupData	1
addGroupParticipant	10
removeGroupParticipant	10
setGroupAdmin	10
removeAdmin	10
setGroupPicture	1
leaveGroup	10
readChat	10
getDeviceInfo	1
checkWhatsapp	10
getAvatar	10
getContacts	1
getContactInfo	1
deleteMessage	10

Метод	Число запросов в секунду
archiveChat	10
unarchiveChat	10
setDisappearingChat	1
sendMessage	50
sendButtons	50
sendTemplateButtons	50
sendListMessage	50
sendLocation	50
sendLink	50
sendContact	50
sendFileByUrl	50
sendPoll	50
clearMessagesQueue	1
showMessagesQueue	1
sendTemplate	50
forwardMessages	50
deleteStatus	10
getIncomingStatuses	1
getOutgoingStatuses	1
getStatusStatistic	1
sendTextStatus	5
sendVoiceStatus	5
sendMediaStatus	5

1.3.13 Рекомендации по интеграции

Рекомендации по интеграции

В данном разделе предоставлены подробные описания рекомендуемых сценариев использования методов с примерами и ответами на часто задаваемые вопросы по написанию интеграции.

[Создание и настройка инстанса](#) [Создание и настройка инстанса с использованием ключа партнёра](#) [Подключение WhatsApp к GREEN API](#) [Отслеживание состояния инстанса](#) [Работа с входящими звонками](#) [Использование хостов GREEN API](#) [Работа с вебхуками и уведомлениями](#)

Создание и настройка инстанса

НАСТРОЙКИ ИНСТАНСА

Рекомендуем делать интеграцию, которая после создания инстанса автоматически устанавливает нужные Вам настройки.

Для получения уведомлений о входящем, исходящем сообщениях, о статусе отправленных сообщений, об изменении состояния авторизации инстанса требуется установить настройки.

Можно использовать [рекомендуемые параметры](#) или расставить их в зависимости от ваших потребностей.

Сценарий создания и настройки инстанса

1. После создания инстанса через [личный кабинет](#), требуется дождаться завершения создания инстанса (создание и инициализация инстанса занимает до 5 минут).
Используя метод [getStateInstance](#), можно проверять состояние инстанса.
Тело ответа `null` при запросе метода [getStateInstance](#), обозначает что инстанс все еще создается. Вы можете использовать данный маркер до момента получения статуса готовности инстанса к работе `notAuthorized`.
2. После получения статуса `notAuthorized` требуется произвести установку настроек методом [SetSettings](#) или через [личный кабинет](#).
3. Установка настроек занимает до 5 минут. В это время можно запросить QR-код для связывания с устройством. Запрос кода рекомендуется делать с использованием [технологии вебсокет](#), так можно гарантировать получение актуального кода на протяжении всего процесса связывания с устройством.
При получении QR-кода через API [требуется делать запросы раз в 2 секунды](#).
4. После успешного связывания с устройством, запросы на получения QR-кода требуется прекратить. Настройки инстанса не меняются со временем, повторно устанавливать настройки инстанса не требуется.
Проверить установку настроек можно, с помощью метода [GetSettings](#).

Рекомендуемые параметры

Рекомендуемые параметры запроса:

```
{
  "webhookUrl": "https://mysite.com/webhook/green-api/", // URL для получения входящих уведомлений
  "webhookUrlToken": "", // токен для доступа к вашему серверу уведомлений
  "delaySendMessagesMilliseconds": 3000, // время отправки сообщений из очереди
  "markIncomingMessagesReaded": "no", // отмечать входящие сообщения прочитанными
  "markIncomingMessagesReadedOnReply": "yes", // отмечать входящие сообщения прочитанными при ответе собеседнику
  "outgoingWebhook": "yes", // получать уведомления о статусах отправленных сообщений
  "outgoingMessageWebhook": "yes", // получать уведомления при отправке с устройства
  "outgoingAPIMessageWebhook": "no", // получать уведомления при отправке с API
  "incomingWebhook": "yes", // получать уведомления о входящих сообщениях
  "deviceWebhook": "no", // получать уведомления об устройстве. Уведомление временно не работает.
  "stateWebhook": "yes", // получать уведомления об изменении состояния авторизации инстанса
  "keepOnlineStatus": "yes", // выставляет статус 'В сети' для вашего аккаунта
  "pollMessageWebhook": "yes", // получать уведомления о создании опроса и голосовании в опросе
  "incomingBlockWebhook": "yes" // получать уведомления о добавлении чата в список заблокированных контактов. Уведомление временно не работает.
}
```

ПОЛУЧЕНИЕ ИНФОРМАЦИИ ОБ АККАУНТЕ

Для получения информации об аккаунте реализованы методы:

1. [GetSettings](#) - предназначен для получения текущих настроек инстанса.
2. [getStateInstance](#) - предназначен для получения состояния инстанса, таких как `notAuthorized`, `authorized` и т.д.
3. [getWaSettings](#) - предназначен для получения информации об аккаунте WhatsApp. Возвращает аватар, номер телефона аккаунта WhatsApp и состояние инстанса.

Создание и настройка инстанса с использованием ключа партнёра

1. Для создания инстанса используется метод партнера [createInstance](#), при создании инстанса требуется передавать настройки инстанса.
2. После создания инстанса требуется каждые 5 секунд непрерывно запрашивать статус созданного инстанса методом [getStateInstance](#), до получения статуса "Не авторизован".
Тело ответа `null` при запросе метода `getStateInstance`, обозначает что инстанс все еще создается. Вы можете использовать данный маркер до момента получения статуса готовности инстанса к работе `notAuthorized`.
3. После получения статуса `notAuthorized` требуется начать запрашивать QR код, вызывая [метод QR](#) каждые две секунды (чтобы клиент постоянно имел актуальный QR код).
4. После успешного связывания с устройством, запросы на получения QR-кода требуется прекратить. Настройки инстанса не меняются со временем, повторно устанавливать настройки инстанса не требуется.
Проверить установку настроек можно, с помощью метода [GetSettings](#).

[Общие рекомендации](#) по созданию и настройке инстанса.

Дополнительные рекомендации (не обязательные):

1. Дать клиентам возможность авторизации по номеру телефона, с помощью метода [getAuthorizationCode](#), в качестве альтернативы методу авторизации по QR.
2. Для получения актуального QR-кода рекомендуется запрашивать его через websocket соединение с помощью метода [scanqrcode](#).

Подключение WhatsApp к сервису GREEN API

Вы можете подключить номер телефона [WhatsApp](#) к инстансу API в вашей интеграции двумя способами.

"Важно!" Перед получением кода авторизации убедитесь, что инстанс не авторизован. Для деавторизации используйте метод [Logout](#).

ПОДКЛЮЧИТЬ ЧЕРЕЗ QR-КОД

1. Получить QR-код через [websocket](#).

- Создайте файл в формате HTML;
- Добавьте в него код из [примера](#);

Соединение устанавливается кодом на JavaScript.

- Создайте локальный хостинг страницы;
- В поля вставьте `apiUrl`, `idInstance` и `apiTokenInstance`;

Где найти `apiUrl`, `idInstance` и `apiTokenInstance` можете узнать в [статье](#).

- Нажмите на кнопку `Get QR`. QR-код появится на экране.

При получении QR-кода через `websocket`, на кнопку `Get QR` можно нажать несколько раз и вам всегда будет возвращаться актуальное изображение QR-кода.

2. Через браузер.

- Получить QR-код можно прямо через поисковую строку браузера.

```
https://qr.green-api.com/waInstance1101*****/0654934b277d4ef693b01ab073d9cd2e97f02db9a532*****
```

3. Через метод [QR](#).

Пример кода.

- Добавьте `apiUrl`, `idInstance` и `apiTokenInstance` из [личного кабинета](#);
- Поле `message` вернёт строку с QR-кодом в кодировке `base64`;
- Чтобы увидеть изображение с QR-кодом, в строке поиска браузера поставьте `data:image/png;base64, {message}`.

"Внимание!" Для успешной авторизации через [QR-код](#), нужно отсканировать его в течение 20 секунд. Иначе QR-код обновиться на стороне WhatsApp и всю процедуру нужно будет начать сначала.

ПРИВЯЗАТЬ ПО НОМЕРУ ТЕЛЕФОНА

Пример кода.

```
POST {{apiUrl}}/waInstance{{idInstance}}/getAuthorizationCode/{{apiTokenInstance}}
```

- Отправьте POST запрос методом [getAutoriationCode](#).
- Добавьте `apiUrl`, `idInstance` и `apiTokenInstance` из [личного кабинета](#);
- Настроить тело запроса - ввести в поле `phoneNumber` номер телефона в международном формате.

Не используйте плюс, нули, скобки или дефисы при вводе номера телефона в [международном формате](#). Например: [70001234567@c.us](#).

- Поле `code` в теле ответа покажет восьмизначный код, который необходимо ввести в приложении WhatsApp.

"Примечание" У способа привязки по номеру телефона нет ограничений по времени для авторизации. Код легко можно отправить другому человеку по почте или в мессенджере. Для авторизации по номеру телефона не требуется камера. Подробнее можете прочитать в нашей [статье](#).

Отслеживание состояние инстанса

Получение состояния инстанса можно реализовать несколькими способами:

1. Использовать метод [GetStateInstance](#) - запрашивать состояние инстанса каждую минуту и в случаи если состояние инстанса изменится выполнять действия:
 - `authorized` - информировать оператора, что инстанс находится в рабочем состоянии;
 - `notAuthorized` - информировать оператора о необходимости повторно связать аккаунт с инстансом и прекращать запросы методов отправки к АПИ (сообщения не будут доставлены, они будут добавляться в очередь и храниться там в течении 24 часов);
 - `blocked` - информировать оператора о необходимости сделать запрос на разблокировку аккаунта и прекращать запросы методов отправки к АПИ (сообщения не будут доставлены, они будут добавляться в очередь и храниться там в течении 24 часов);
 - `starting` - сделать перезапуск инстанса методом [reboot](#), подождать 2 минуты и повторить получение состояния инстанса, если статус инстанса не изменился, то информировать оператора что в данный момент инстанс находится в режиме запуска и возможна задержка при отправке сообщений;
 - `yellowCard` - информировать оператора о получении предупреждения на аккаунте, требуется выполнить инструкции указанные с статье [yellowCard](#). Сообщения отправленные после получения статуса хранятся в очереди к отправке 24 часа. Для продолжения работы инстанса требуется сделать [перезагрузку](#) инстанса.
2. Получение состояния инстанса с использованием уведомлений.
Для получения уведомлений Вам необходимо:
 - Включить настройку "Получать уведомления об изменении состояния авторизации аккаунта" (методом [setSettings](#) или через [личный кабинет](#)).
 - Сделать обработчик уведомлений [статуса инстанса](#) и при получении одного из статусов выполнять действия указанные в предыдущем пункте.

"Примечание" - Рекомендуем использовать систему основанную на методе [GetStateInstance](#), при использовании получения инстанса уведомлениями, дать возможность оператору вручную проверить статус инстанса (по кнопке) или использовать оба метода одновременно.

- Рекомендуем самостоятельно реализовать систему оповещения при изменении состояния инстанса на статусы `'notAuthorized'`, `'blocked'`, `'yellowCard'` (отправлять письмо на почту, отправлять уведомление в Telegram или WhatsApp).

Работа с входящими звонками

В зависимости от потребностей интеграции, работу с [уведомлениями о входящих звонках](#) можно реализовать несколькими способами:

1. Если Вам не нужна работа с входящими звонками, то рекомендуем делать сброс входящего уведомления [incomingCall](#)
 - При работе обработчика уведомлений - установить удаление неизвестных типов уведомлений по полю `typeWebhook`. Так все ненужные вам уведомления будут удаляться из очереди.
 - При работе через [HTTP API](#) - удалять уведомления методом [deleteNotification](#).
 - При работе через [Webhook Endpoint](#) - отдавать статус код 200.
2. Если Вам необходимо получать только факт входящего звонка, можно сделать обработчик на получение уведомления `typeWebhook: incomingCall` и отслеживать статус `status: offer`. Все остальные статусы исключать из обработки звонков.
3. Если вам необходимо видеть только результат звонка и не информировать оператора о поступлении входящего звонка, тогда требуется отслеживать и записывать в чат все статусы кроме `offer`. Также можно сократить информацию по неотвеченным звонкам.

Например:

- При получении уведомления (`typeWebhook: incomingCall + status: offer`) - просто сбрасывать уведомление.
 - При получении других статусов - записывать в чат с собеседником записи:
 - `pickUp` - "Отвеченный звонок"
 - `hungUp` - "Пропущенный звонок"
 - `missed` - "Пропущенный звонок"
 - `declined` - "Пропущенный звонок"
4. Если необходимо получать все статусы звонков, требуется реализовать обработку всех уведомлений типа [incomingCall](#). Необходимо при получении уведомления транслировать в чат запись о звонке.

Например:

- Получаем входящий звонок (`typeWebhook: incomingCall + status: offer`), записываем в чат `Входящий вызов` и информируем оператора.
- Завершение звонка (по причинам: ответа оператора, сброса звонка, истечения времени ожидания) - получаем уведомление (`typeWebhook: incomingCall`) и в зависимости от значения поля `status` делаем запись в чат:
 - `pickUp` - "Отвеченный звонок"
 - `hungUp` - "Сброшенный звонок" - оператор сбросил звонок или в системе установлена настройка или на телефоне активирована функция `Не беспокоить` (положили трубку)
 - `missed` - "Пропущенный звонок" - собеседник завершил звонок
 - `declined` - "Пропущенный звонок" - собеседник не дождался ответа

Получение имени звонящего собеседника

1. Получить уведомление [incomingCall](#) о входящем звонке, в поле `from` указан номер аккаунта собеседника.
2. Использовать метод [GetContactInfo](#) для определения информации о звонящем собеседнике.
3. Использовать поля `name` или `contactName` из метода [GetContactInfo](#) для получения имени собеседника.

Использование хостов GREEN API

Для выполнения запросов HTTP API на сервера GREEN API требуется использовать параметры указанные в настройках Вашего инстанса в [личном кабинете](#):

- **apiUrl** - ссылка на хост GREEN API для выполнения методов
- **mediaUrl** - ссылка на хост GREEN API для отправки файлов (методы [UploadFile](#), [SendFileByUpload](#))
- **idInstance** - уникальный номер Вашего инстанса
- **apiTokenInstance** - ключ доступа Вашего инстанса

При выполнении запросов с параметрами из [личного кабинета](#) гарантирует минимальное время выполнения запроса и высокую вероятность успешного выполнения запроса.

1. [Изменение хоста на партнерском аккаунте](#)
2. [Как плавно перевести инстансы на выделенные сервера](#)
3. [Влияние смены хоста на работу интеграции](#)

1. ИЗМЕНЕНИЕ ХОСТА НА ПАРТНЕРСКОМ АККАУНТЕ

В случае перевода партнерского аккаунта на работу на выделенных серверах, рекомендуется постепенно перевести все текущие инстансы на новые сервера. Для этого достаточно удалить старый инстанс и создать новый. Перевод на выделенные сервера осуществляется для более стабильной работы системы.

При переходе на выделенные сервера временно можно использовать универсальный хост для работы старых и новых инстансов.

- Универсальный хост: <https://api.green-api.com>
- Универсальный медиа-хост для отправки файлов: <https://media.green-api.com>

В дальнейшем при переходе на новые инстансы на выделенных серверах, следует изменить хост на указанный в личном кабинете, для уменьшения времени ответа АПИ и уменьшения вероятности появления проблем с доступностью.

2. КАК ПЛАВНО ПЕРЕВЕСТИ ИНСТАНСЫ НА ВЫДЕЛЕННЫЕ СЕРВЕРА

Ограничения по срокам перехода на новые инстансы на выделенных серверах отсутствуют.

При возникновении необходимости сменить инстанс можно использовать моменты, когда инстанс становится неавторизованным, например:

1. Когда клиент отвязывает устройство
2. Если клиент переходит с тарифа Бизнес на партнерскую тарификацию, при удалении инстансов Бизнес
3. Когда инстансы [разлогиниваются](#) автоматически;

Следует сообщить клиентам о необходимости перепривязать устройство для более стабильной работы.

Вы можете отслеживать статус инстанса через получение [уведомлений](#) при включенной настройке stateWebhook или используя метод [GetStateInstance](#).

3. ВЛИЯНИЕ СМЕНЫ ХОСТОВ НА РАБОТУ ИНТЕГРАЦИИ

Использование **не предназначенного для инстанса хоста** может сломать работу интеграции, требуется при работе в переходный период временно использовать универсальные хосты.

Перевод инстансов на выделенные сервера и смена хостов при указании временного хоста и переключении на рекомендуемый, происходит незаметно для клиентов и не влияет на работу интеграции.

Работа с вебхуками и уведомлениями

Получение входящих данных в GREEN API реализовано через входящие уведомления. Есть 2 способа получать уведомления [HTTP API](#) и [Webhook Endpoint](#).

Ниже приведены рекомендации по работе с вебхуками и уведомлениями:

1. [Получение входящих уведомлений](#)
2. [Обработка входящих уведомлений](#)
3. [Безопасность при получении уведомлений](#)
4. [Логирование и обработка ошибок](#)
5. [Масштабирование и распределение нагрузки](#)

1. ПОЛУЧЕНИЕ ВХОДЯЩИХ УВЕДОМЛЕНИЙ

Работа через HTTP API

При работе через методы HTTP API требуется вызывать последовательно 2 метода API:

1. Вызываем метод [receiveNotification](#) в зависимости от наличия сообщений, настроек метода и настроек инстанса в ответе Вы получите объект с номером и объектом уведомления.
Если уведомлений в очереди нет, в ответе придет null.
2. После получения уведомления необходимо обработать входящее уведомление обработчиком и удалить уведомление методом [deleteNotification](#) передав номер этого уведомления.
Если метод завершился статусом кодом 500 и более требуется повторить запрос метода.

Работа через Webhook Endpoint

При работе через методы Webhook Endpoint требуется настроить веб-сервер, способный принимать HTTP запросы.

Веб-сервер должен:

1. Принять входящий [POST запрос](#) (вебхук)
2. Обработать принятое уведомление
3. Вернуть ответ с кодом состояния 200 (после получения от сервера статус кода 200 уведомление будет удалено из очереди).
Время ожидания ответа от Вашего веб-сервера составляет 180 секунд, если сервер не отдаст статус код 200, сервер GREEN API сделает паузу в 60 секунд и повторит отправку текущего уведомления.

2. ОБРАБОТКА ВХОДЯЩИХ УВЕДОМЛЕНИЙ

Все типы и форматы уведомлений указаны в [документации](#).

Рекомендуется сделать обработку входящих уведомлений согласно инструкции ниже:

1. После получения уведомления сервисом обработчика, желательно проверить схему уведомления (JSONSchema).

Не рекомендуется делать проверку схемы на дополнительные поля, так как API может расширяться добавляя новые функции и информацию.

Если Вы получили уведомление в котором отсутствует поле или его формат не верный, требуется зафиксировать такое уведомление в системе логирования и проинформировать оперетора указав данное уведомление, это поможет зафиксировать и разобраться в ошибке в будущем.

2. Сервис обработчика должен найти обязательное поле `typeWebhook` и в зависимости от его типа обработать или сбросить уведомление.
3. После обработки уведомления обработчик должен отдать статус код 200.

Так все не обрабатываемые вам уведомления будут удаляться из очереди и последующие уведомления будут приходить корректно.

Если Вы не используете некоторые типы уведомлений, то рекомендуем делать сброс не нужных уведомлений, отдавая статус код 200, так Вы исключите ошибки обработчика и при выходе нового функционала у Вас не возникнет проблем в работе интеграции.

Пример простого веб-сервера от GREEN API:

- [Python](#) с инструкцией по [установке](#).
- [Golang](#) с инструкцией по [установке](#).
- [NodeJS](#) с инструкцией по [установке](#).

3. БЕЗОПАСНОСТЬ ПРИ ПОЛУЧЕНИИ УВЕДОМЛЕНИЙ

При проектировании сервиса необходимо уделить внимание безопасности.

При работе с технологией [HTTP API](#) Ваш IP адрес закрыт и не доступен другим пользователем интернета, данная технология хороша работает в небольших системах с малым объемом данных.

При работе с технологией [Webhook Endpoint](#) Ваш IP адрес доступен другим пользователем, требуется исключить вредоносные воздействия на Вашу систему извне.

Рекомендации по безопасности при использовании технологии [Webhook Endpoint](#):

1. Настройте сетевой экран (Firewall), разрешите доступ к серверу только с требуемых Вам IP адресов.
2. Ограничение доступа только с определённых IP адресов. Вы можете настроить Ваш веб-сервер таким образом, чтобы он принимал входящие запросы только с [IP адресов GREEN API](#).
3. Использовать токен авторизации при отправке уведомлений. Укажите токен авторизации в поле `webhookUrlToken` через [личный кабинет](#) или методом установки настроек [setSettings](#).

"Пример кода фильтрации IP адресов на python"

```

'''python
from flask import Flask, request

from datetime import datetime
import json

app = Flask(__name__)

# Список IP-адресов Green Api

ALLOWED_IPS = [
    "51.250.84.44",
    "51.250.94.65",
    "51.250.89.177",
    "64.226.125.75",
    "158.160.49.84",
    "46.101.109.139",
    "51.250.95.149",
    "51.250.12.167",
    "209.38.202.24"]
'''

```

```
# Функция для проверки разрешенных IP-адресов

def check_ip():
    client_ip = request.remote_addr
    if client_ip not in ALLOWED_IPS:
        return False
    return True

@app.before_request
def before_request():
    if not check_ip():
        return "Access Denied", 403

# Остальной код вашего приложения...

if __name__ == "__main__":
    app.run()
...
```

4. ЛОГИРОВАНИЕ И ОБРАБОТКА ОШИБОК

Рекомендуется настроить систему логирования:

1. Используете многоуровневую настраиваемую систему логирования. Логируйте проблемы схемы на уровне ERROR, новые типы уведомлений на уровне WARN, необработываемые типы сообщений на уровне DEBUG и работу системы на уровне INFO. Включая нужный уровень логирования, Вы сможете отлаживать работу системы и определять все проблемы, связанные с интеграцией.
2. Рекомендуется сообщать оператору о неизвестных типах уведомлений и об ошибках уведомлений, это позволит максимально быстро локализовать проблему и даст больше информации оператору, не пропустив еще неподдерживаемые типы уведомлений интеграцией.

Обязательно логируйте данные по уведомлениям, в записи должна содержаться информация включая дату и время получения - `timestamp`, тип уведомления - `typeWebhook` и остальные данные уведомления. Это позволит вам проводить анализ работы, при неожиданных сценариях работы системы.

5. МАСШТАБИРОВАНИЕ И РАСПРЕДЕЛЕНИЕ НАГРУЗКИ

При увеличении размера системы Вы можете столкнуться с проблемой масштабирования. Рекомендуется масштабировать систему, распределяя нагрузку на несколько микросервисов обработчиков. Для этого можно использовать балансировщик нагрузки.

В качестве примера балансировщика приведем веб-сервер `nginx`, Вы можете использовать несколько экземпляров вашего сервера и распределять нагрузку между ними, используя `nginx`.

[Nginx](#) - это высокопроизводительный веб-сервер, который также может использоваться в качестве балансировщика нагрузки (процесс равномерного распределения веб-трафика между несколькими серверами), предотвращая перегрузку одного сервера. Nginx применяет различные алгоритмы для оптимального распределения трафика и может быть настроен для обеспечения надежности системы при выходе из строя одного или нескольких серверов.

Воспользуйтесь одним из приведённых способов, чтобы развернуть несколько экземпляров сервера:

Ручное развертывание на разных серверах или виртуальных машинах:

- Создайте или арендуйте несколько физических серверов, виртуальных машин или контейнеров.
- Настройте каждый сервер так, чтобы он содержал копию вашего приложения.

Использование контейнеризации с Docker:

- Создайте Docker-образ вашего приложения.
- Запустите несколько контейнеров с этим образом на одном или нескольких хост-машинах.

Использование облачных провайдеров:

- Разверните несколько экземпляров вашего сервера на платформе облачных провайдеров, таких как Amazon Web Services (AWS), Google Cloud Platform (GCP) или Microsoft Azure.
- Используйте инструменты управления облачной инфраструктурой для автоматизации развертывания и масштабирования серверов.

"Пример файла конфигурации nginx для балансировщика нагрузки"

```
...
http {
    upstream app{
        server 10.2.0.100;
        server 10.2.0.101;
        server 10.2.0.102;
    }

    // Этот сервер принимает весь трафик на порт 80 и передает его вышестоящему потоку.
    // Обратите внимание, что имя вышестоящего потока и proxy_pass должны совпадать.

    server {
        listen 80;

        server_name mydomain.com;

        location / {
            include proxy_params;

            proxy_pass http://app;

            proxy_redirect off;
            proxy_http_version 1.1;
            proxy_set_header Upgrade $http_upgrade;
            proxy_set_header Connection "upgrade";
        }
    }
    ...
}
```

1.4 Личный кабинет

1.4.1 Личный кабинет

Регистрация

Зарегистрируйтесь в личном кабинете для доступа к инстансам и функциям API

1.4.2 Регистрация в личном кабинете

Для доступа к инстансам, токену и функциям API необходимо пройти регистрацию в [личном кабинете](#) пользователя.

Чтобы зарегистрироваться, следует нажать на кнопку **Войти** в правом верхнем углу на главной странице сайта.

GREEN API Документация SDK Чат-боты Стать партнером ★ Интеграции Релизы Блог Новости FAQ **Войти** Русский

WhatsApp API

Отправляйте и получайте сообщения, картинки и видео через стабильный шлюз **WhatsApp API**

Бесплатный инструмент для разработчиков

Зелёный свет Вашим Идеям!

Работать бесплатно

Стабильно **Безопасно** **Быстро**

Перед Вами откроется поле регистрации.




Регистрация

* Электронная почта:

* Пароль:



* Повторите пароль:

* Страна:  Россия ▼

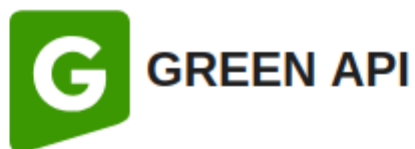
Регистрируясь в Green-API, вы соглашаетесь с нашей политикой конфиденциальности и условиями использования

[Зарегистрироваться](#)

Уже есть аккаунт Green-API? [Войти](#)

 Русский ▼ 

Если этого не произойдёт, для перехода на поле регистрации можно воспользоваться кнопкой [Регистрация](#) в нижней части формы [Вход](#).



Вход

Запомнить меня[Забыли пароль?](#)

Войти

[Зарегистрироваться](#)

[▶ Видеоинструкция](#)

[🌐 Русский](#) ▼

В форме регистрации требуется:

1. Указать электронную почту;
2. Придумать пароль и повторить придуманный пароль для подтверждения. Его лучше записать или сохранить в менеджере пароля. Он потребуется для входа в личный кабинет в будущем;
3. Указать страну регистрации;
4. Поставить галочку о прочтении пользовательского соглашения (ознакомиться с ним можно кликнув по словосочетанию [Пользовательское соглашение](#))
5. Нажать на кнопку [Зарегистрироваться](#) .

В форме регистрации, в самом низу, также можно выбрать язык, на котором будет отображаться интерфейс сайта.

После прохождения регистрации Вы сможете войти в личный кабинет. Для повторного входа можно воспользоваться кнопкой [Войти](#) , на главной странице сайта, о которой было написано выше. Больше не потребуется регистрироваться заново. Необходимо заполнить только форму с названием [Вход](#) .



Вход

 Запомнить меня [Забыли пароль?](#)

[Зарегистрироваться](#)
[Видеоинструкция](#)
Русский

Для повторного входа укажите:

1. Электронную почту, которую указывали при регистрации;
2. Пароль, который Вы придумали при регистрации;
3. Можно поставить галочку в пункте `Запомнить меня`, если хотите сохранить электронную почту и пароль для автоматического входа при следующем посещении личного кабинета;
4. Нажмите на кнопку `Войти`.

Если Вы забыли пароль, его можно обновить через кнопку `Забыли пароль?`. В таком случае откроется форма `Восстановление доступа`.



Восстановление доступа

Для восстановления доступа укажите адрес электронной почты вашей учетной записи.

Восстановить доступ

[Вернуться назад](#)

Внутри формы необходимо:

1. Указать электронную почту, которая использовалась для регистрации;
2. Нажать на кнопку [Восстановить доступ](#).

Инструкция для восстановления доступа придёт на указанную электронную почту. Внутри письма можно будет увидеть временный пароль. Его необходимо скопировать и пройти процедуру входа в личный кабинет через поле [Вход](#), как было описано ранее, но в поле [Пароль](#) указать пароль из письма, пришедшего на почту.

Сгенерированный пароль из письма можно поменять на свой собственный.

Изменить пароль

* Текущий пароль:

* Новый пароль:

* Новый пароль еще раз:

Изменить пароль

Разлогинить аккаунт (Выйти)

Выполняет выход из личного кабинета

Версия: 0.0.186

Настройки пользователя

Страна:  Россия ▾

Сохранить настройки

Для этого в личном кабинете следует перейти во вкладку [Профиль](#).

1. В графе [Текущий пароль](#) указать пароль из письма;
2. В графе [Новый пароль](#) указать тот пароль, который Вы придумаете самостоятельно или вам предложит менеджер паролей браузера;
3. Повторно введите Ваш новый пароль ещё раз в графе ниже;
4. Нажмите на кнопку [Изменить пароль](#).

Из действующего аккаунта можно выйти в графе [Профиль](#) по кнопке [Разлогинить аккаунт \(Выйти\)](#).