Введение. Виды форм. Принцип работы.

> Теоретическая часть для разработчика сайта (разработчика HTML-формы) и для разработчика бота.

Веб-формы расширяют канал коммуникаций с ботом и позволяют в текстовом боте использовать все возможности HTML и JavaScript, таким образом бот, по функционалу становится полноценной заменой любому веб-сайту. Бот теперь не ограничивает пользователя в коммуникациях только обменом текстом, аудио или файлами.

> Актуальный исходный код веб-формы реализующий все три вида форм смотрите по ссылке: go-to-the-mars.html

> Пример работы веб-формы приведенной выше смотрите в Telegram боте https://t.me/metabot_test_form_bot

Разработка веб-формы, расширяющей диалоговую коммуникацию происходит в несколько шагов:

1. Перед началом разработки необходимо выбрать один из вариантов реализации формы и согласовать с разработчиком веб-формы и разработчиком чат-бота (если форма и чат-бот разрабатываются параллельно). Также необходимо согласовать формат обмена данными.
   - Универсальная в виде ссылки (для любого мессенджера).
   - WebApp, на основе inline-кнопки (только для Telegram).
   - WebApp, на основе keyboard-кнопки (только для Telegram).
2. После того как вы определились с видом формы, необходимо разработать HTML-форму, разместить его на вашем сервере, подключить ее к backend вашего сайта (для обработки AJAX / API запросов), чтобы backend мог принять данные с формы и отправить их в API в Metabot.

>

> **Для разработчика формы:** реализация любого вида формы не имеет принципиальных архитектурных отличий, одна и также веб-форма может использоваться для любого вида формы путем добавления не сложных разветвлений в JS-коде веб-формы. Отличие только в том, что вариант с keyboad-кнопкой не требует дополнительного слоя для backend, но при этом имеет ряд ограничений, подробнее см в описании этого вида формы ниже. Мы рекомендуем использовать вариант Web App с inline-кнопокй + универсальную форму для любого мессенджера.

> **Для разработчика бота:** см отдельную страницу документации для реализации необходимого вида формы в боте.

Дополнительное изучение возможностей работы Telegram Web Apps.

Универсальная форма в виде ссылки

> Ключевая особенность: универсальный вариант, подходит для любого мессенджера.

Форма открывается в отдельной странице браузера. Как обычная страница web-браузера.

> **Для разработчика бота:** вы можете добавить в боте условия, проверяя текущий мессенджер лида, чтобы открывать для нативного Telegram-канала форму с помощью inline кнопки, а для других каналов отправлять ссылку на форму в виде сообщения. Учтите, что в других мессенджерах нельзя удалять сообщения, поэтому заранее предусмотрите время жизни хэш-кода лида который указывается в ссылке на форму

Для Telegram ссылку можно отправить в виде inline-кнопки, но это все равно будет не "Web App приложение", а обычная страница открытая в браузере.

Если в мессенджере открытие ссылок во внешнем браузере выключено то форма откроется в браузере мессенджера, но все равно работа с данным видом формы будет отличаться от формы в виде Web App для inline/keyboard кнопки. При этом пользователю все равно доступна возможность позволяющая открыть ссылку во внешнем браузере. После закрытия формы нет гарантий, что пользователь вернется в браузер, поэтому желательно прислать пользователю уведомление в бота, чтобы он нажал на него и вернулся в бота.

Если вы планируете использовать формы только в Telegram, то можете пропустить данный вид формы и перейти к описанию формы на основе Web App, открывающейся с помощью inline-кнопки. Но, желательно понимать отличия и изучить весь раздел документации.

Принцип работы, по шагам:

1. Бот генерирует ссылку с уникальным хэш-кодом, привязанным к лиду.
2. Бот отправляет эту ссылку в мессенджер в виде обычного сообщения.
3. Пользователь бота нажимает на ссылку, открывается браузер с формой.
4. Пользователь бота заполняет форму и нажимает кнопку для отправки формы.
5. Форма отправляет данные на backend сайта, где размещена форма.
6. Backend сайта отправляет данные формы в API Metabot, в данные должен быть включен уникальный хэш лида.
7. Metabot принимает и сохраняет данные заполненной формы .
8. Если все ок, то страница с формой в браузере должна быть закрыта, это делается с помощью простого JS кода (подробности рассматриваются на отдельной странице документации с описанием создания HTML-формы).

Пример JS кода для открытия ссылки на Telegram бота и закрытия страницы

```
<script>
  function closeForm() {
      location.href="tg://resolve?domain=metabot_test_form_bot";
      window.close();
  }
</script>
```

После заполнения формы, данные введенные пользователем необходимо отправить на бэк вашего сайта, а затем в Metabot API используя токен бота (чтобы токен бота не фигурировал в коде HTML-формы). Далее страницу браузера необходимо автоматически закрыть, с помощью JS-кода встраиваемого в HTML-форму. Рекомендуется отправлять данные на бэк с помощью REST API, для того чтобы после отправки не выполнять дополнительный рендеринг страницы (чтобы потом ее просто закрыть). Т.е. отправляем данные на бэк, убеждаемся что ответ от API - 200, т.е. все ОК и сразу закрываем форму, пользователь возвращается в бота. При этом желательно учесть в коде HTML-формы варианты, если что-то пошло не так, или пользователь остался на форме, а хэш код лида для формы истек (если вы генерируете токен со сроком действия), в этих случаях можно вывести сообщение о том что форма устарела, что пользователю необходимо вернуться в бота и запросить форму повторно, также дополнительно можно уведомить Metabot по API, чтобы бот автоматически выслал ссылку на новую форму.

Есть нюанс: перед закрытием формы можно вызвать открытие ссылки для перехода в мессенджер (на случай если после закрытия страницы не будет открывается мессенджер). Но, для PC, например, открытие ссылки с Telegram ботом не срабатывает автоматически и у пользователя остается открытой другая вкладка в браузере (если у него включено открытие ссылок мессенджера во внешнем браузере и в браузере открыто несколько вкладок).

> Ссылка может быть отправлена в бота в виде простого текста или inline-кнопки Telegram, содержащей ссылку на форму. Но в любом случае, для рассматриваемого варианта формы, нажатие на ссылку или кнопку приведет к открытию странице в

> браузере и после заполнения формы страницу необходимо автоматически закрывать. Здесь речь идет именно об обычной inline-кнопке с ссылкой для Telegram, а не о Telegram WebApp-форме, на основе inline-кнопки которая открывает форму в Web App (во всплывающем окне, на котором расположен WebView). Описание вариантов форм с WebApp смотрите ниже.

Информация для разработчика бота:

Отличие между Inline Web App Button и обычной inline кнопки в виде ссылки заключается в том, что Inline Web App Button формируется передачей параметров web_app и url а для обычная inline-кнопка в виде ссылки просто передачей параметра url. Детали можно найти в описании API Telegram.

https://core.telegram.org/bots/webapps#inline-button-web-apps
https://core.telegram.org/bots/api#inlinekeyboardbutton
https://core.telegram.org/bots/api#inlinekeyboardmarkup
https://core.telegram.org/bots/api#sendmessage

Inline кнопка с ссылкой для Telegram реализуется на основе JS Callback команды бота и метода bot.sendMessage или с помощью плагина Common.TelegramComponents.MenuHelper.sendMessage с передачей дополнительных параметров для работы inline-кнопки как ссылки.

Отличие bot.sendMessage от плагина Common.TelegramComponents.MenuHelper.sendMessage в том что функция плагина сама внутри вызывает bot.sendMessage, и при этом автоматически сохраняет ID последних сообщений в атрибутах лида. Этот атрибут нужен для использования в коде JS Callback, а также маршруте бота, если требуется удаление кнопки-ссылки на форму или самого сообщения с ссылкой. Функционал хранения идентичен работе с Фото-слайдером для Telegram (описано в документации по работе с командой JS Callback и файлами Telegram).

Команда JS callback необходима для обработки нажатия других кнопок которые могут быть отправлены вместе с кнопкой-ссылкой на форму или для fallback, если форму не заполнили, но отправили любой текст в мессенджер.

Команда бота JS Callback не должна в данном случае обрабатывать событие отправки формы. Т.к. событие отлавливается с помощью Internal API Endpoint. В JS Callback можно отлавливать событие заполнения формы только для формы реализованной на основе keyboad-кнопки в Telegram.

WebApp-форма, на основе inline-кнопки

> Данный вид формы работает только в Telegram.

> Рекомендуемый вид формы для Telegram бота.

> Ключевая особенность: форма открывается в всплывающем окне на котором расположен WebView (встроенный браузер).

Отличие данного варианта от универсальной формы в виде ссылки, в том, что форма работает как Web App, т.е. это специальный режиме браузера встроенного в Telegram. Этот режим гарантирует возврат в бота после заполнения формы, а также не позволит открыть ссылку во внешнем браузере (а также включает дополнительный функционал для взаимодействия веб-страницы с ботом). В обычном же режиме для универсальной формы, нет гарантий, что когда веб-страница будет закрыта, пользователя перекинет в бота.

Принцип работы, по шагам:

1. Бот генерирует ссылку с уникальным хэш-кодом, привязанным к лиду.
2. Бот отправляет эту ссылку в виде inline-кнопки в Telegram-мессенджер.
3. Пользователь бота нажимает на кнопку, открывается Web App с формой (т.е. пользователь остается в Telegram, в мессенджере открывается встроенный браузер).
4. Пользователь бота заполняет форму и нажимает кнопку для отправки формы.
5. Форма отправляет данные на backend сайта, где размещена форма.
6. Backend сайта отправляет данные формы в API Metabot, в данные должен быть включен уникальный хэш лида.
7. Metabot принимает и сохраняет данные заполненной формы.
8. Если все ок, то страница с формой в браузере должна быть закрыта, это делается с помощью простого JS кода (подробности рассматриваются на отдельной странице документации с описанием создания HTML-формы).

WebApp-форма, на основе keyboard-кнопки

> Данный вид формы работает только в Telegram.

>

> Альтернативный вид формы для бота в Telegram. Используется, если для вас крайне затратно реализовать дополнительный backend-слой для отправки данных в Metabot API

> Имеет ряд ограничений и неудобств (список смотрите ниже)

> Ключевая особенность: форма открывается в всплывающем окне на котором расположен WebView (встроенный браузер). Для отправки данных в Metabot API не требуется дополнительный backend.

Для пользователя форма работает аналогично виду формы на основе inline-кнопки, но только кнопка для открытия формы расположена в нижней клавиатуре, в так называемой "кнопке-калькулятора".

Отличие данного варианта от универсальной формы в виде ссылки, в том, что форма работает как Web App, т.е. это специальный режиме браузера встроенного в Telegram. Этот режим гарантирует возврат в бота после заполнения формы, а также не позволит открыть ссылку во внешнем браузере (а также включает дополнительный функционал для взаимодействия веб-страницы с ботом). В обычном же режиме для универсальной формы, нет гарантий, что когда веб-страница будет закрыта, пользователя перекинет в бота.

Форма открываемая с помощью keyboard-кнопки (так называемой "кнопки-калькулятора"), это упрощенный вариант, чтобы не подключать дополнительный back-end к HTML-форме для пересылки результата сбора данных с HTML-формы в Metabot API. На первый взгляд такой вариант более простой, но имеет ограничения, описанные ниже, поэтому **рекомендуется для Telegram использовать вариант WebApp-формы открываемой с помощью inline-кнопки.**

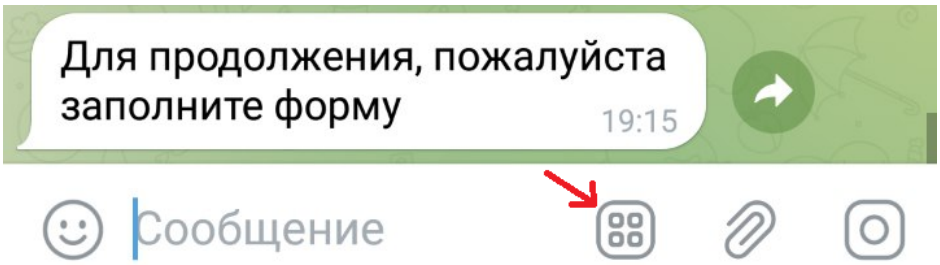
Принцип работы, по шагам:

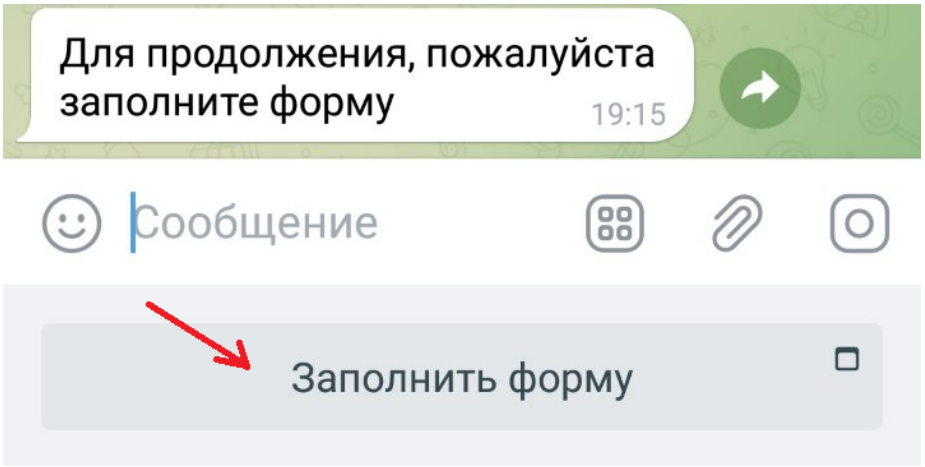
1. Бот генерирует ссылку с уникальным хэш-кодом, привязанным к лиду.
2. Бот отправляет эту ссылку в виде keyboard-кнопки в Telegram-мессенджер ("кнопка-калькулятор" отображаемая в нижней части мессенджера).
3. Пользователь бота нажимает на кнопку, открывается Web App с формой (т.е. пользователь остается в Telegram, в мессенджере открывается встроенный браузер).
4. Пользователь бота заполняет форму и нажимает кнопку для отправки формы.
5. В коде HTML формы размещается JS код, для передачи введенных данных в телеграмм: window.Telegram.WebApp.sendData(JSON.stringify(formData)).
6. Для дополнительной защиты, желательно, в отправляемые данные включить уникальный хэш-код лида.
7. Metabot принимает данные заполненной формы от Telegram и сохраняет эти данные.

8. Страница с формой в браузере будет закрыта автоматически, после выполнения метода Telegram.WebApp.sendData, если же вы не используете sendData, то можно вызвать метод Telegram.WebApp.close().

Ограничения и неудобства данного вида формы:

- Для такого вида формы: максимальный размер данных отправляемых с формы в бота составляет 4096 байт;
- Менее удобный UI/UX в виде кнопки отображаемой в нижней части браузера, также это меняет шаблон поведения пользователя, например, пользователь использовал inline-кнопки в боте, а теперь ему нужно дополнительно отслеживать кнопки "калькулятора";
- Кнопка может скрываться, если пользователь напишет что-то в чат-бот (возможен вариант повторного обновления кнопки, чтобы она не исчезала);
- В некоторых альтернативных мобильных приложениях для Telegram (например Plus Messenger для Android) список keyboard-кнопок может быть изначально схлопнут и необходимо нажимать в нижней части мессенджера на иконку для отображения кнопок меню (см скрины ниже).





Версия #3
Юрий Гарашко создал 2023-06-14 13:53:53 UTC
Ирина Петрова обновил 2025-11-11 07:20:19 UTC