

Выполнить JavaScript Callback

Результатом работы данной команды является прием файлов в Телегам боте, отправленных пользователем.

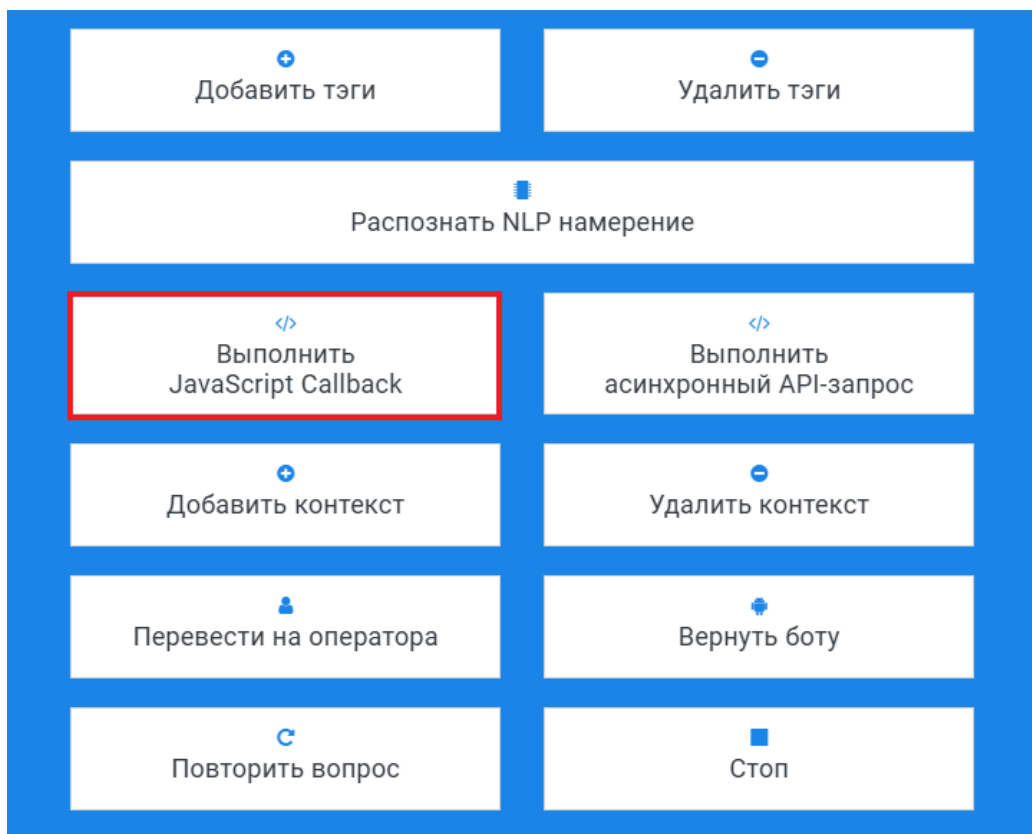
Принцип работы данной команды:

- Когда данная команда запущена, ее состояние за циклируется в текущей точке и текущий и любой следующий входящий вебхук будет обрабатываться этим JavaScript кодом;
- Выйти из команды и перейти к следующей или к другому скрипту можно только с помощью возврата специального значения из JavaScript

Данную команду можно использовать, для:

- Запроса любого значения: текста, файла, фото, медиа;
- Интеграции с любой системой: машиной JSON состояний или любым NLP (за цикливая диалог на NLP), системой обработки голосовых сообщений, системой распознавания изображений.

Добавить в скрипт ее можно, выбрав пункт **Выполнить JavaScript Callback** из списка **Команд**:



После выбора данной команды в списке **Команд** перед вами откроется диалоговое окно настройки свойств:

Добавить команду



Настройте свойства команды

☐ Использовать условие

Тип команды:

Выполнить JavaScript Callback	Browse
-------------------------------	--------

Сообщение-вопрос отправляемое до старта данной команды:

Прикрепите фото.

Сообщение при возникновении ошибки:

Фото имеет неподходящий формат.

☒ Запустить Callback сразу при старте этой команды

JavaScript Callback:

<div></div>

Создать

С пунктом **Использовать условие** вы можете познакомиться на странице [Условие выполнения команды JS](#).

Следующие поля являются обязательными:

- **Сообщение-вопрос отправляемое до старта данной команды** — сообщение которое будет отправлено в мессенджер при старте данной команды, поле обязательно для заполнения ТОЛЬКО, если выключено поле **Запустить Callback сразу при старте этой команды**;
- **Сообщение при возникновении ошибки** — сообщение которое будет отправлено в мессенджер при возникновении ошибки;
- **Запустить Callback сразу при старте этой команды** — если поле неактивно, то JavaScript код запустится только после того как пользователь получит **Сообщение-вопрос отправляемое до старта данной команды** в мессенджер и что-то

ОТВЕТИТ;

Если поле активно и это первый запуск **JavaScript Callback** данной команды, то в **JavaScript** будет установлена в **true** системная предопределенная переменная **isFirstImmediateCall**, во всех других случаях **isFirstImmediateCall** устанавливается в **false**. Это необходимо, чтобы можно было проинициализировать компонент и, например, выслать текст вопроса или отправить изначально меню и т.д.

- **JavaScript Callback** — здесь указывается исходный код вашего JavaScript компонента.

По кнопке **Создать** как итог команда появится в редакторе скрипта:

Выполнить JavaScript Callback

Сообщение-вопрос отправляемое до старта данной команды:
Прикрепите фото.

Сообщение при возникновении ошибки:
Фото имеет неподходящий формат.

Запустить Callback сразу при старте этой команды: Да

JavaScript Callback:

```
const { imageUploader } = require("Common.TelegramComponents.ImageUploader")
imageUploader.setMaxFileSizeMb(2)
imageUploader.setMinImageWidth(0)
imageUploader.setMinImageHeight(0)
imageUploader.setScriptCodeForFilesIsNotSupported("files_is_not_supported")
imageUploader.setFilesCustomCaptionName("person_files")
imageUploader.setBtnCancelCaption("Отмена")
imageUploader.setBtnOkCaption("Отправить")
imageUploader.calcPromptMessage = function() {
  let maxFileSizeMb = this.getMaxFileSizeMb()
  let minImageWidth = this.getMinImageWidth()
  let minImageHeight = this.getMinImageHeight()
  let msg = "\bОтправьте свое фото\b" +
    (maxFileSizeMb > 0 ? ("r\nr\nr\nРазмер фото: не более " + maxFileSizeMb + "
Mb</i>" : "") +
    (minImageWidth > 0 ? ("r\nШирина: не менее " + minImageWidth + " px</i>" : "") +
    (minImageHeight > 0 ? ("r\nВысота: не менее " + minImageHeight + " px</i>" : "")
  return msg
}
imageUploader.calcPromptKeyboard = function() {
  return [
    [
      {
        "text": this.calcBtnCancelCaption(),
        "callback_data": "btn_cancel",
        "text_button_label": this.getBtnCancelNoInlineLabelCode()
      },
    ],
  ]
}
return imageUploader.getComponentResult()
```

Возвращаемые значения

JavaScript Callback всегда должен возвращать данные с помощью «**return**» в определенном формате:

Для завершения инициализации

- **return null.**

Используется, чтобы платформа ничего не делала со своей стороны, ожидая ответа пользователя. Вместо **retrurn null** можно ничего не указывать после ключевого слова **return**.

Для прерывания цикла

- **return true;**
- **return { "result": true ... другие параметры };**
- **return { "break": true ... другие параметры }.**

Используется для выхода из замыкания и переходу к следующей команде текущего скрипта (если не указан параметр для перехода к другому скрипту).

Прерывание с помощью **throw new Error("break")** недопустимо, в таком случае JS прервется, но из заикливания не выйдет, т.е. будет распознано как **return null**.

Под "**другими параметрами**" здесь подразумевается, что можно указать параметры в JSON-объекте, на которые платформа умеет реагировать. На данный момент для **"result": true**, можно указывать следующие параметры:

- **"run_script_by_code": "код_вашего скрипта"** — если указан этот параметр, то выполняется выход из замыкания и переход к указанному скрипту по коду.

Для повторения цикла

- **return false;**
- **{ "result": false ... другие параметры }.**

Используется для завершения обработки JavaScript (выхода из JavaScript), без прерывания заикливания. Callback продолжит выполнение цикла и при приеме любого входящего сообщения мы опять попадем в JavaScript код данной команды (если не указан параметр для перехода к другому скрипту).

Под "**другими параметрами**" здесь подразумевается, что можно указать параметры в JSON-объекте, на которые платформа умеет реагировать. На данный момент для **"result": false**, можно указывать следующие параметры:

- **"fallback": "текст вашего фолбэка"** — используется для вывода вместе с фолбэком текста-объяснения;
- **"run_script_by_code": "код_вашего скрипта"** — используется для запуска другого скрипта по коду, если параметр передан, то fallback и повтор вопроса выводиться не будут;
- **"is_repeat_prompt": true / false** — выводить или нет повтор вопроса. Повтор вопроса выводится только, если он указан в настройках команды.

Версия #5

Юрий Гарашко создал 12 June 2023 10:59:04

Ирина Петрова обновил 18 December 2024 06:31:07