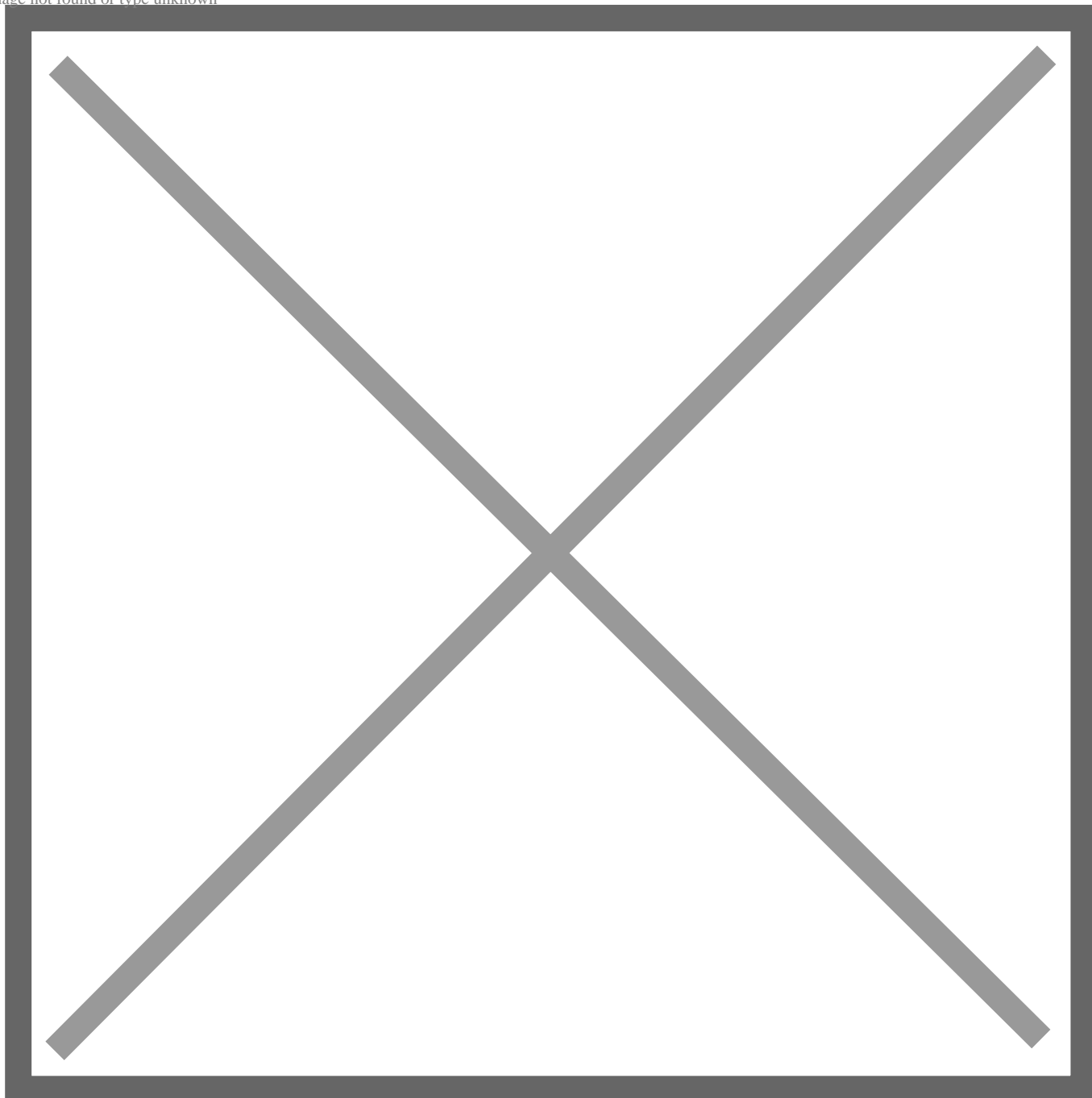


Подключение распознавания файлов в различных каналах

Редактирование настроек канала Metabot Widget

В канале **Metabot Widget** через редактирование необходимо:

Image not found or type unknown

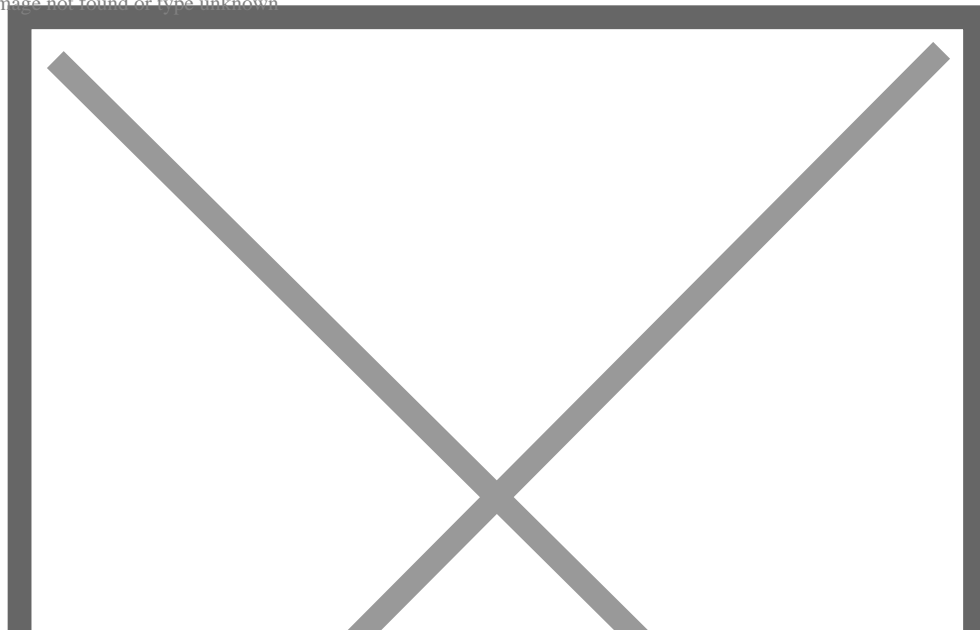


1. Поставить активность чекбокса **Разрешить отправку файлов**.
2. В поле **Список поддерживаемых расширений файлов** добавить список разрешенных форматов файлов:

pdf,doc,docx,xls,xlsx,csv,txt,png,jpe,jpg,jpeg,gif,bmp,ico,svg,svgz,tif,tiff,jfif,ai,drw,pct,psd,raw,webp,xbm,dib,pgp,apng,pjpeg,avif,avi,divx,flv,m4v,mkv,mov,mp4,mpeg,mpg,ogm,ogv,ogx,rm,rmvb,smil,webm,wmv,xvid,3gp,3g2,qt,asx

3. Выбрать **Только меню** в поле **Реакция на файлы (документы)**.
4. Сохранить настройки.

Image not found or type unknown



Редактирование настроек скрипта с типом Fallback

В название скрипта Fallback рекомендуем добавить “+ ФАЙЛЫ” для оперативного поиска в списке скриптов. Необходимо проверить корректность настроек NLP в скрипте.

Затем заполните скрипт командами:

- **Первой командой** должна идти команда **Выполнить JavaScript**, которая получает из вебхука событие: есть ли в сообщении с распознанным текстом файл и добавляет переменную **is_attachment**, что файл есть;

```
let whJob = bot.getWebhookJob()
let eventType = whJob.event_type

memory.setAttr('is_attachment', false)
if (eventType === 'user_attachments') {
  memory.setAttr('is_attachment', true)
  bot.disableRepeatMessageText() // погасит вывод сообщения с повтором вопроса
  bot.hideRepeatMessageButtons() // погасит вывод меню с повтором вопроса
}
```

- **Второй командой** должна идти команда **Выполнить JavaScript**, которая фиксирует сам файл в хранилище и формирует ссылку для передачи в последствии в сообщениях/в письме на почту;

```
// Библиотека с помощью которой можно записать в карочку лида данные не отправляя их в тг
let TelegramMessage = require('Common.Integrations.Telegram')
let msg = new TelegramMessage()

// Все входящие файлы
let attachments = bot.getAllAttachments()

if( Boolean( attachments?.[0]?.url )){

uploadData = bot.downloadFileFromUrl( attachments[0].url)
msg.debug(' Пользователь отправил файл: ' + uploadData.url)

lead.setAttr(' file', uploadData.url)
// Запуск скрипта...
bot.runScriptByCodeForLead("recieveFile", lead.getData(' id' ))

bot.stop()
}
```

- **Третьей командой** должна идти команда **Выполнить JavaScript**, которая срабатывает в случае, когда файла в сообщениях **нет** (проверка в условии). JavaScript подбирает случайный ответ из списка, фиксирует дату сообщения, заносит в Google таблицу нераспознанный текст;

```
// Условие

return !(memory.getAttr(' is_attachment' ) *1)
```

```
// JavaScript

var randomStrings = [
    "Не совсем понимаю, о чём вы.",
    "Извините, я вас не понял. Перефразируйте, пожалуйста.",
    "Попробуйте воспользоваться разделами меню или сказать другими словами.",
    "Простите, не понял вас, попробуйте снова."
];

randomIndex = Math.ceil(( Math.random() * randomStrings.length - 1 ));

lead.setAttr(' randomAnswer', randomStrings[randomIndex]);
```

```

var ДатаСообщения = new Date(Date.now());

var ДатаСообщенияСтр =
    ДатаСообщения.getFullYear() + "-" +
    ("0" + (ДатаСообщения.getMonth()+1)).slice(-2) + "-" +
    ("0" + ДатаСообщения.getDate()).slice(-2);

var GoogleSheetsService = require('Common.Integrations.GoogleSheets') // Плагин для работы с
Google Sheets

GoogleSheetsService.sheetId = '1pLbxTJG2YI6LghJ4FMCCBkQ7ybTL0Y7ZGw4yvaGCsQo' // ID
скопированный из таблицы
GoogleSheetsService.listName = 'Рexay.Про' // Название листа

// Параметры где ключ - название столбца, значение - данные которые занесутся в строку
let params = {
    "Bot ID": botId,
    "Лид ID": leadId,
    "Время": ДатаСообщения,
    "Сообщение": bot.getIncomingMessage(),
}

let result = GoogleSheetsService.addRow(params) // Функция для добавления строк в таблицу

debug(result)

```

- **Четвертой командой** должна идти команда **Отправить текст** с рандомным текстом, которая срабатывает в случае, когда файла в сообщениях **нет** (проверка в условии);

```

// Условие

return !(memory.getAttr('is_attachment')*1)

```

Текст команды

```
{{ $randomAnswer }}
```

- **Пятой командой** должна идти команда **Отправить текст** с текстом, которая срабатывает в случае, когда в сообщении **есть** файл (проверка в условии);

```
// Условие
```

```
return (memory.getAttr('is_attachment')*1)
```

Текст команды

Файл принят, передадим на изучения группе поддержки. С вами свяжутся либо через чат-бота, либо через имеющиеся контакты.

А пока что можете воспользоваться автоматическим функционалом чат-бота.

- При необходимости **шестой командой** должна идти команда **Email**, которая срабатывает в случае, когда в сообщении **есть** файл (проверка в условии). На почту отправляется письмо с файлом пользователя;

```
// Условие
```

```
return (memory.getAttr('is_attachment')*1)
```

Получатель	email, на который необходимо отправить письмо. ({{ \$bot.botSupportMails }})
Тема	*Название бота поддержки*: лид отправил в бота файл
Содержимое	В чат-боте пользователь прислал в бота файл ID клиента для поиска в диалогах на платформе Metabot: {{ &\$leadId }} Ссылка на файл: {{ \$is_attachment }} С уважением, Ваш Metabot

При необходимости, можно создать системный атрибут с ключом **botSupportMails**, типом **variable** и значением равным всем почтам, на которые должно поступить письмо.

- **Последней командой** должна идти команда **Повторить вопрос**.

Версия #1

Ирина Петрова создал 23 April 2024 12:47:32

Ирина Петрова обновил 23 April 2024 13:11:41