

ВК: Плагин по форматированию текста и кнопок

????????? ???????	VK
??????????????	?????????????? ??????? ?? Metabot
?????	???? ?????
???? ?????????	30 ??? 2026
?????????? ????? ???????????	31 ??? 2026

Описание

Плагин позволяет форматировать текст и кнопки отправляемые в ВК.

Подключение

```
snippet(' Business. Integrations. VK' )  
var msg = new VkMessage()
```

Использование

Текст

```
msg.text = ' Привет'  
msg.send()
```

Кнопки

DSL:

- `==` - новый ряд
- `[]` - кнопки в одном ряду
- `Текст[команда]` - кнопка
- `Сайт[https://...]` - ссылка

Подпись обрезается до 20 символов, кнопок не больше 7.

`callback_data` всегда `{ command: "..."}.`

```
msg.text = ' Меню'
msg.keyboard = ' Да[ yes]==Нет[ no] '
msg.send()
return true
```

В VK кнопка часто приходит **текстом подписи**, не `callback`.

```
snippet(' Business. Integrations. VK' )
var msg = new VkMessage()
var MENU = ' Да[ yes]==Нет[ no] '

msg.keyboard = MENU
var p = msg.getMessagePayload()

if (!isFirstImmediateCall && p.callback_data && p.callback_data !== 'undefined') {
    bot.runScriptByCodeForLead(p.callback_data, leadId)
    return true
}

msg.text = ' Выберите: '
msg.send()
return true
```

Также можно прописывать номера кнопок: `1`, `2`

Картинки и файлы (отправка)

Плагин принимает только `image` и `file`. Файл должен быть указан как публичный URL.

```
msg.text = 'Фото'
msg.filesData = [
  { type: 'image', url: 'https://example.com/pic.jpg' }
]
msg.send()
```

```
msg.text = 'Документ'
msg.filesData = [
  { type: 'file', url: 'https://example.com/doc.pdf' }
]
msg.send()
```

После `send()`:

- `lead.getJsonAttr('vk_last_send_result')`
- `lead.getAttr('vk_last_message_id')`

Входящие вложения

```
msg.getImages()
msg.getFiles()
msg.getAllAttachments()
```

Видео из VK сейчас принимать нельзя

Исходный код плагина

```
/**
 * Upload в VK: только image и file (photo/doc).
 * Текст+кнопки с вложением платформа для VK отправляет отдельно (не как TG/Max).
 */
if (typeof VkMessage === 'undefined') {
  var vkNormalizeCommand = function (data) {
    var s
    var o

    if (data == null) {
      return ''
    }
  }
```

```

if (typeof data === 'object') {
    return String(data.command || data.callback_data || '')
}

s = String(data)
if (s.charAt(0) === '{') {
    try {
        o = JSON.parse(s)
        return String(o.command || o.callback_data || '')
    } catch (e) {}
}

if (s.indexOf('btn_static_') === 0) {
    return s.slice('btn_static_'.length)
}

return s
}

var vkMakeCallback = function (command) {
    return { command: String(command) }
}

var vkSafeCall = function (fn, args) {
    try {
        if (typeof fn !== 'function') {
            return []
        }
        return fn.apply(bot, args || [] || [])
    } catch (e) {
        return []
    }
}

var vkMapAttachmentType = function (raw) {
    var t = String(raw || 'file').toLowerCase()
    if (t === 'photo' || t === 'image' || t === 'sticker') {
        return 'image'
    }
    if (t === 'document' || t === 'doc' || t === 'file') {
        return 'file'
    }
}

```

```

    }
    return null
}

VkMessage = function () {
    this._text = null
    this._keyboard = null
    this._attachments = []
    this._answer_symbol = '>'
    this._max_button_label = 20
    this._max_buttons = 7
    this._baseUrl = 'https://app.metabot24.com/' /*меняем на сервер который будет
использоваться для отправки сообщений*/
    this.webhookPayload = bot.getWebhookPayload()
}

Object.defineProperty(VkMessage.prototype, 'text', {
    get: function () {
        return this._text
    },
    set: function (value) {
        this._text = value
    }
})

Object.defineProperty(VkMessage.prototype, 'keyboard', {
    get: function () {
        return this._keyboard
    },
    set: function (value) {
        this._keyboard = value ? this.createVkKeyboard(value) : null
    }
})

Object.defineProperty(VkMessage.prototype, 'answer_symbol', {
    set: function (value) {
        this._answer_symbol = value
    }
})

Object.defineProperty(VkMessage.prototype, 'filesData', {

```

```

get: function () {
    return this._attachments
},
set: function (value) {
    var i
    var prepared
    var item
    var list

    this._attachments = []
    if (!value) {
        return
    }
    if (Object.prototype.toString.call(value) !== '[object Array]') {
        this.debug('filesData must be an array')
        return
    }

    list = value
    for (i = 0; i < list.length; i++) {
        item = list[i]
        prepared = this.prepareAttachment(item)
        if (prepared) {
            this._attachments.push(prepared)
        }
    }
}

}))

Object.defineProperty(VkMessage.prototype, 'baseServerUrl', {
    set: function (value) {
        if (value) {
            this._baseServerUrl = String(value)
            if (this._baseServerUrl.charAt(this._baseServerUrl.length - 1) !== '/') {
                this._baseServerUrl += '/'
            }
        }
    }
})

```

```

VkMessage.prototype.createVkKeyboard = function (keyboard) {

```

```

var rows = String(keyboard).split('==')
var result = []
var count = 0
var i
var j
var cols
var row
var item
var btn
var code

for (i = 0; i < rows.length; i++) {
    cols = rows[i].split('||')
    row = []

    for (j = 0; j < cols.length; j++) {
        item = String(cols[j] || '').trim()
        if (!item) {
            continue
        }
        if (count >= this._max_buttons) {
            break
        }

        btn = null
        if (/https?:\/\//.test(item) && /\[[^\[\]]+\]/.test(item)) {
            code = item.match(/\[[^\[\]]+\]/)[0].slice(1, -1)
            btn = {
                text: this._clip(item.replace(/\[[^\[\]]+\]/g, '')),
                url: code,
                callback_data: vkMakeCallback(code),
                text_button_label: String(count + 1)
            }
        } else if (/^\[[^\[\]]+\]/.test(item)) {
            code = item.match(/\[[^\[\]]+\]/)[0].slice(1, -1)
            btn = {
                text: this._clip(item.replace(/\[[^\[\]]+\]/g, '')),
                callback_data: vkMakeCallback(code),
                text_button_label: String(count + 1)
            }
        } else {

```

```

        btn = {
            text: this._clip(item),
            callback_data: vkMakeCallback(item),
            text_button_label: String(count + 1)
        }
    }

    row.push(btn)
    count++
}

if (row.length) {
    result.push(row)
}

return result
}

VkMessage.prototype._clip = function (label) {
    var text = String(label || '').trim()
    if (text.length > this._max_button_label) {
        return text.slice(0, this._max_button_label)
    }
    return text
}

VkMessage.prototype.debug = function (data) {
    bot.sendMessage(
        'VK debug\n\n' + (typeof data === 'string' ? data : JSON.stringify(data))
    )
}

VkMessage.prototype.processLocalFile = function (filePath) {
    var attrs
    var businessId

    attrs = bot.getDebugAttributes ? bot.getDebugAttributes() : null
    businessId = attrs && attrs.business_id != null ? attrs.business_id : ''
    return this._baseServerUrl + 'uploads/business/' + businessId + '/' + filePath
}

```

```

/**
 * Готовит вложение для bot.sendMessage(..., attachments).
 * Поддерживаемые type: image, file (photo как image, document/doc как file).
 */
VkMessage.prototype.prepareAttachment = function (item) {
    var type
    var url
    var out

    if (!item || typeof item !== 'object') {
        return null
    }

    type = vkMapAttachmentType(item.type || item.file_type)
    if (!type) {
        this.debug({
            error: 'unsupported attachment type for VK upload',
            type: item.type || item.file_type,
            hint: 'use image or file only'
        })
        return null
    }

    url = item.url || item.file_url || item.file_id || null
    if (!url && item.file_path) {
        url = this.processLocalFile(item.file_path)
    }
    if (!url) {
        this.debug({
            error: 'attachment url missing',
            item: item
        })
        return null
    }

    out = {
        type: type,
        url: String(url)
    }
    if (item.caption !== null) {

```

```

    out.caption = String(item.caption)
  } else if (item.text != null) {
    out.caption = String(item.text)
  }
  if (item.payload && typeof item.payload === 'object') {
    out.payload = item.payload
  }
  if (item.file_id && item.file_id !== url) {
    out.file_id = item.file_id
  }
  return out
}

VkMessage.prototype.GetFiles = function (mimeInclude, mimeExclude) {
  if (mimeExclude === undefined) {
    return vkSafeCall(bot.GetFiles, [mimeInclude])
  }
  return vkSafeCall(bot.GetFiles, [mimeInclude, mimeExclude])
}

VkMessage.prototype.getImages = function (mimeInclude) {
  if (mimeInclude === undefined) {
    return vkSafeCall(bot.getImages, [])
  }
  return vkSafeCall(bot.getImages, [mimeInclude])
}

VkMessage.prototype.getVideos = function (mimeInclude) {
  if (mimeInclude === undefined) {
    return vkSafeCall(bot.getVideos, [])
  }
  return vkSafeCall(bot.getVideos, [mimeInclude])
}

VkMessage.prototype.getAllAttachments = function (filesMimeExclude) {
  if (filesMimeExclude === undefined) {
    return vkSafeCall(bot.getAllAttachments, [])
  }
  return vkSafeCall(bot.getAllAttachments, [filesMimeExclude])
}

```

```

VkMessage.prototype._resolveMessageId = function (resultPayload) {
    var messageId = null

    if (typeof bot.getLastMessageId === 'function') {
        try {
            messageId = bot.getLastMessageId()
        } catch (e) {}
    }

    if (messageId !== null) {
        return messageId
    }

    if (!resultPayload) {
        return null
    }

    if (resultPayload.response !== null) {
        return resultPayload.response
    }

    if (resultPayload.message_id !== null) {
        return resultPayload.message_id
    }

    if (resultPayload.result !== null && typeof resultPayload.result !== 'object') {
        return resultPayload.result
    }

    if (resultPayload.result && resultPayload.result.message_id !== null) {
        return resultPayload.result.message_id
    }

    return null
}

VkMessage.prototype.send = function () {
    var text
    var sent
    var resultPayload
    var messageId
    var hasAttachments
    var keyboard

    hasAttachments =

```

```

Object.prototype.toString.call(this._attachments) === '[object Array]' &&
this._attachments.length > 0

if (!this._text && !hasAttachments) {
    this.debug('text or filesData is required')
    return false
}

text = this._text != null && String(this._text) !== '' ? String(this._text) : ''
keyboard = this._keyboard && this._keyboard.length ? this._keyboard : null

if (hasAttachments) {
    sent = bot.sendMessage(text, keyboard, this._attachments)
} else if (keyboard) {
    sent = bot.sendMessage(text, keyboard)
} else {
    sent = bot.sendMessage(text)
}

resultPayload = bot.getLastResultPayload()
lead.setJsonAttr('vk_last_send_result', {
    sent: sent,
    has_keyboard: !!keyboard,
    has_attachments: hasAttachments,
    attachments_count: hasAttachments ? this._attachments.length : 0,
    result: resultPayload
})

messageId = this._resolveMessageId(resultPayload)
if (messageId != null) {
    lead.setAttr('vk_last_message_id', String(messageId))
}

if (sent === false || (resultPayload && resultPayload.error)) {
    this.debug({
        error: 'send failed',
        sent: sent,
        has_attachments: hasAttachments,
        result: resultPayload
    })
}

```

```

    return this.getMessagePayload()
}

VkMessage.prototype.addReplyToText = function () {
    var payload = this.getMessagePayload()
    if (payload.input_type === 'press' && payload.text) {
        this._text = this._text + '\n\n' + this._answer_symbol + ' ' + payload.text
    }
    this._keyboard = null
    this._attachments = []
    return this.send()
}

VkMessage.prototype._attachIncoming = function (result) {
    var attachments
    var hasText

    attachments = this.getAllAttachments()
    if (Object.prototype.toString.call(attachments) !== '[object Array]') {
        attachments = []
    }
    result.attachments = attachments

    hasText = result.text !== null && String(result.text) !== '' && result.text !== 'undefined'
    if (attachments.length && !hasText && result.input_type !== 'press') {
        result.input_type = 'attachment'
        if (!result.text || result.text === 'undefined') {
            result.text = ''
        }
    }

    return result
}

VkMessage.prototype.getMessagePayload = function () {
    var result = {}
    var payload = this.webhookPayload || {}
    var data
    var mid
    var eventObj

```

```

var raw
var incoming
var r
var b
var row
var btn
var cmd

if (payload.callback_query) {
  data = payload.callback_query.data
  mid = null
  if (payload.callback_query.message) {
    mid = payload.callback_query.message.message_id
  }
  cmd = vkNormalizeCommand(data)
  result.message_id = mid || lead.getAttr('vk_last_message_id') || null
  result.callback_data = cmd
  result.text = this._buttonLabel(cmd) || cmd
  result.input_type = 'press'
  result.payload = payload
  return this._attachIncoming(result)
}

eventObj = payload.object || payload
if (
  payload.type === 'message_event' ||
  eventObj.type === 'message_event' ||
  (eventObj.event_id !== null && eventObj.payload !== null)
) {
  raw = eventObj.payload
  cmd = vkNormalizeCommand(raw)
  result.message_id =
    eventObj.conversation_message_id || lead.getAttr('vk_last_message_id') || null
  result.callback_data = cmd
  result.text = this._buttonLabel(cmd) || cmd
  result.input_type = 'press'
  result.payload = payload
  return this._attachIncoming(result)
}

incoming =

```

```

(payload.message && payload.message.text) ||
(payload.object && payload.object.message && payload.object.message.text) ||
(payload.object && payload.object.text) ||
bot.getIncomingMessage()

if (incoming != null && String(incoming) !== '') {
    result.message_id =
        (payload.message && payload.message.id) ||
        (payload.object && payload.object.message && payload.object.message.id) ||
        (payload.object && payload.object.conversation_message_id) ||
        null
    result.text = String(incoming)
    result.input_type = 'write'
    result.payload = payload

    if (Object.prototype.toString.call(this._keyboard) === '[object Array]') {
        for (r = 0; r < this._keyboard.length; r++) {
            row = this._keyboard[r]
            for (b = 0; b < row.length; b++) {
                btn = row[b]
                if (!btn || !btn.callback_data) {
                    continue
                }
                // VK часто шлёт подпись кнопки текстом; также принимаем номер 1/2/...
                if (
                    btn.text === result.text ||
                    String(btn.text_button_label || '') === result.text
                ) {
                    result.callback_data = vkNormalizeCommand(btn.callback_data)
                    result.input_type = 'press'
                    return this._attachIncoming(result)
                }
            }
        }
    }

    return this._attachIncoming(result)
}

result.message_id =
    (payload.message && payload.message.id) ||

```

```

    (payload.object && payload.object.message && payload.object.message.id) ||
    (payload.object && payload.object.conversation_message_id) ||
    null
result.text = 'undefined'
result.input_type = 'undefined'
result.callback_data = 'undefined'
result.payload = payload
return this._attachIncoming(result)
}

```

```

VkMessage.prototype._buttonLabel = function (command) {

```

```

    var needle
    var r
    var b
    var row
    var btn
    var cb

```

```

    if (
        Object.prototype.toString.call(this._keyboard) !== '[object Array]' ||
        command == null
    ) {
        return null
    }

```

```

    needle = vkNormalizeCommand(command)
    for (r = 0; r < this._keyboard.length; r++) {
        row = this._keyboard[r]
        for (b = 0; b < row.length; b++) {
            btn = row[b]
            cb = vkNormalizeCommand(btn.callback_data)
            if (cb === needle) {
                return btn.text
            }
        }
    }
    return null

```

```

}
}

```

Версия #1

Ирина Петрова создал 26 August 2026 10:44:00

Ирина Петрова обновил 26 August 2026 11:00:39