Instantly share code, notes, and snippets.

Birdie0 / **ifttt-webhooks-extended-guide.md**

Last active 7 months ago

Code Revisions 18 Stars 271 Forks 43

How to use Discord Webhooks

ifttt-webhooks-extended-guide.md

⚠️ This gist is no longer updated! For maintained, improved and even more extended guide click here.

How to use Discord Webhook

It's a JSON

First, learn JSON. It's not programming language, not even close. Just follow syntax rules and you will be fine.

Recommended sources:

- Learn X in Y minutes | Where X=json
- JSON official site

Structure of Webhook Body

Second, learn the structure. All root elements are *optional* unless stated otherwise. Body have to include `content` or `embeds`, otherwise request will fail.

> `element : [data type]` - description

- `[string]` - text
  - `[website url]` - ex.: `https://google.com`

        - `[image url]` - ex.: `https://www.w3schools.com/html/pic_mountain.jpg`

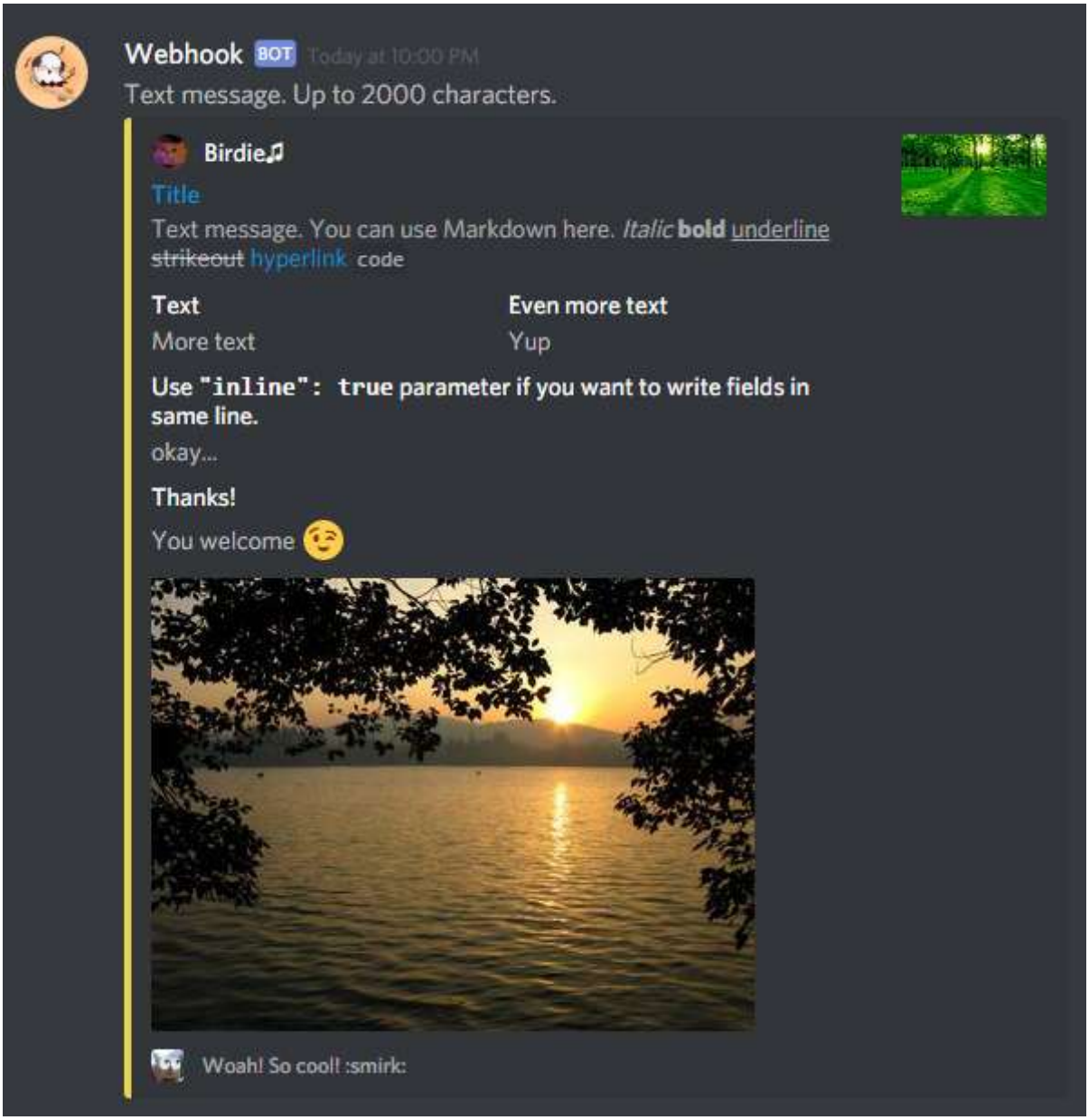
- `[array]` - comma-separated elements. ex.: `[1, 2, 3, 4]`
- `[number]` - ex. `1337` , `420.69` , `-1000` , etc.
- `[boolean]` - can be `true` or `false` only.
- `[object]` - can include `key: value` s
- `username` : `[string]` - overrides the default username of the webhook
- `avatar_url` : `[image url]` - overrides the default avatar of the webhook
- `content` : `[string]` - simple message, the message contains (up to 2000 characters)
- `embeds` : `[array]` - array of embed objects. That means, you can use more than one in the same body
    - `author` : `[object]` - embed author object
        - `name` : `[string]` - name of author
        - `url` : `[website url]` - url of author. If `name` was used, it becomes a hyperlink
        - `icon_url` : `[image url]` - url of author icon
    - `title` : `[string]` - title of embed
    - `url` : `[website url]` - url of embed. If `title` was used, it becomes hyperlink
    - `description` : `[string]` - description text
    - `color` : `[number]` - color code of the embed. You have to use Decimal numeral system, not Hexadecimal. Use color picker
    - `fields` : `[array]` - array of embed field objects
        - `name` : `[string]` - name of the field
        - `value` : `[string]` - value of the field
        - `inline` : `[boolean]` - if true, fields will be displayed in same line, but there can only be 3 max in same line or 2 max if you used thumbnail
    - `thumbnail` : `[object]` - embed thumbnail object
        - `url` : `[image url]` - url of thumbnail
    - `image` : `[object]` - embed image object
        - `url` : `[image url]` - image url
    - `footer` : `[object]` - embed footer object
        - `text` : `[string]` - footer text, doesn't support Markdown
        - `icon_url` : `[image url]` - url of footer icon

Example for a webhook

```
{
  "username": "Webhook",
  "avatar_url": "https://i.imgur.com/4M34hi2.png",
  "content": "Text message. Up to 2000 characters.",
  "embeds": [
    {
      "author": {
        "name": "Birdie♫",
        "url": "https://www.reddit.com/r/cats/",
        "icon_url": "https://i.imgur.com/R66g1Pe.jpg"
      },
      "title": "Title",
      "url": "https://google.com/",
      "description": "Text message. You can use Markdown here. *Italic* **bold** __u
      "color": 15258703,
      "fields": [
        {
          "name": "Text",
          "value": "More text",
          "inline": true
        },
        {
          "name": "Even more text",
          "value": "Yup",
          "inline": true
        },
        {
          "name": "Use `\"inline\": true` parameter, if you want to display fields i
          "value": "okay..."
        },
        {
          "name": "Thanks!",
          "value": "You're welcome :wink:"
        }
      ],
      "thumbnail": {
        "url": "https://upload.wikimedia.org/wikipedia/commons/3/38/4-Nature-Wallpap
      },
      "image": {
        "url": "https://upload.wikimedia.org/wikipedia/commons/5/5a/A_picture_from_C
      },
      "footer": {
        "text": "Woah! So cool! :smirk:",
        "icon_url": "https://i.imgur.com/fKL31aD.jpg"
      }
    }
```

```
    ]
}
```

And how it looks



Getting Started

Account on IFTTT

Visit IFTTT and create an account (if you haven't one).

Webhook on Discord

1. Go to **Server settings -> Webhooks -> Create Webhook**
2. Setup name, avatar and the channel, where it will be posted. Copy *Webhook URL*. **Do not share! Very dangerous!**
3. Click `Save` and then the `Done` button

Creating an Applet

If `this`

1. Go to My Applets -> `New Applet`
2. Click `[+]this`
3. `Choose a service`. In theory, you could use all of them
4. `Choose the trigger`. Read the description below every trigger and choose the needed one
5. `Complete trigger fields`. There can be more than one step. Read the descriptions and examples

Then `that`

1. Click `[+]that`
2. `Choose the action service`. You need `Webhooks`. Use the search bar
3. `Choose action`. Choose `Make a web request`
4. Paste your *Webhook URL* to **URL** field
5. Select `POST` method
6. Select `application/json` content type
7. And now the hardest part™. You need to create your JSON body. Follow the structure, use it as an example and don't forget about common sense. Press `+ Ingredient` button and put the *Ingredients* into appropriate fields. If something says `URL` put it in `"url":"{{URL}}"`, if something says `ImageURL`, try to put that into `"image": {"url":"{{ImageURL}}"}`. There's no universal solution
8. Click `Create Action` and then `Finish`
9. Done!

Tips

- Use this awesome website to generate valid json wirh built-in preview: Discohook, also you can send webhooks to your server with it if you just wan't fancy embed in your channel without any automatization.

- Don't forget to check your JSON body with a JSON validator. If you don't know any use one of these:
  - JSON Formatter
  - JSONLint
  - JSON Editor Online
- If the webhook *doesn't work*, check log for errors. **My Applets** -> *choose applet* -> *click gear* -> ***View activity log***. `Maker error` means your JSON body has errors.
- Discord has built-in embeds for Twitter, Youtube and other sites so you can just add the link to the webhook: `{"content": "{{Link}}"}` .

Examples

Check https://birdie0.github.io/discord-webhooks-guide Examples section!

Load earlier comments...

Birdie0 commented on Mar 5, 2021 • edited Author

@rajada1

> But is there any way to get message.id via api?

You need bot for that, like use method that returns array of messages from channel etc.

rajada1 commented on Mar 5, 2021

> @rajada1
>
> > Mas há alguma maneira de conseguir message.id via api?
>
> Você precisa de bot para isso, como usar método que retorna matriz de mensagens do canal etc.

Thanks !

blackdackota commented on Mar 22, 2021 • edited

So I learned some HTML and Python the last time I messaged you and I'm working on building the bot for what I talked to you about last time, (turns out you don't need to login to see standings). I ran into a problem though I'm not sure how to solve. When I scrape the page I don't get the full code from the website.

This is my code:

```
from bs4 import BeautifulSoup
import requests

html_text = requests.get('https://app.playvs.com/league/eastern-regional-high-school-rocket-
league-rec?redirect=https://www.playvs.com/virginia&text=VA').text
soup = BeautifulSoup(html_text, 'lxml')
print(html_text)
```

This is the output:

<!doctype html>

```
<link rel="icon" type="image/png" sizes="32x32" href="/favicon-32x32.png">
<link rel="icon" type="image/png" sizes="16x16" href="/favicon-16x16.png">
<link rel="stylesheet" href="/static/css/bundle.2f8661ce.css">
```

<script> window.__APOLLO_STATE__ = {}; window.env = {"COOKIE_DOMAIN":".playvs.com","IS_BROWSER":true,"IS_NODE_ENV_PRODUCTION":true,"IS_NODE_ENV_DEVELOPMENT":false,"NODE_ENV":"production","MAINTENANCE":undefined,"API_ENV":"production","API_URL":"https:\u002F\u002Fapi.playvs.com","BATTLEDOTNET_AUTH_URI":"https:\u002F\u002Fapi.playvs.com\u002Fauth\u002Fbattle-dot-net\u002Foauth\u002FauthorizeNoRedirect","DISCORD_AUTH_URI":"https:\u002F\u002Fapi.playvs.com\u002Fpartner_api\u002Fdiscord\u002Fv2\u002FauthNoRedirect","DISCORD_SCHOOL_AUTH_URI":"https:\u002F\u002Fapi.playvs.com\u002Fpartner_api\u002Fschooldiscord\u002Fv2\u002FauthNoRedirect","EPICGAMES_AUTH_URI":"https:\u002F\u002Fapi.playvs.com\u002Fauth\u002Fepic-games\u002Foauth\u002FauthorizeNoRedirect","XBOXLIVE_AUTH_URI":"https:\u002F\u002Fapi.playvs.com\u002Fauth\u002Fxbox-live\u002Foauth\u002FauthorizeNoRedirect","FACEBOOK_AUTH_URI":"https:\u002F\u002Fapi.playvs.com\u002Fauth\u002Ffacebook-rally","FORTNITE_AUTH_URI":"https:\u002F\u002Fapi.playvs.com\u002Fauth\u002Fepic-games\u002Foauth\u002FauthorizeNoRedirect","GRAPHQL_URI":"https:\u002F\u002Fapi.playvs.com\u002Fgraphql","RIOT_AUTH_URI":"https:\u002F\u002Fapi.playvs.com\u002Fauth\u002Friot\u002Foauth\u002FauthorizeNoRedirect","STEAM_AUTH_URI":"https:\u002F\u002Fapi.playvs.com\u002Fauth\u002Fsteam\u002FnoRedirect","TWITCH_AUTH_URI":"https:\u002F\u002Fapi.playvs.com\u002Fauth\u002Ftwitch\u002Foauth\u002FauthorizeNoRedirect","TWITCH_SCHOOL_AUTH_URI":"https:\u002F\u002Fapi.playvs.com\u002Fauth\u002Ftwitchschool\u002Foauth\u002FauthorizeNoRedirect","FACEBOOK_PIXEL_ID":undefined,"INTERCOM_KEY":undefined,"LINKEDIN_PIXEL_ID":undefined,"SHEERID_PROGRAM_ID":"5ea217969318bb1c35c428e1","SEGMENT_KEY":"KxR2CiH74fpRKPizTaAd0sX9n7ea2hFh","SENTRY_DSN":"https:\u002F\u002F6affc91010eb4d8d901efb96564091af@sentry.io\u002F1519550","SNAPCHAT_PIXEL_ID":undefined,"STRIPE_PUBLIC_KEY":"pk_live_zRgvwgznx8Tpwhaje7lSX6CV","STRIPE_SECRET_KEY":"sk_test_0fsotpaV5U4MP5CeNGRfiEIH","ZENDESK_KEY":"987e7048-

4533-4f91-bf9e-0c323ef166c7","FORTNITE_CONNECT_FEATURE":true,"SCRIMMAGE_FEATURE":true,"FEATURE_SCRIMMAGE_FILTERS":true,"PROJECT_SPAWN":true,"HIDE_TEAMS":false,"SETTLE_MATCH_FEATURE":true,"PLAYER_PROFILES":false,"PLAYER_STAT_HISTORY":false,"PLAYER_STANDINGS":true,"OPEN_ENROLLMENT":false,"REC_VARSITY_FEATURE":true,"PLAYER_LED_TEAMS_FEATURE":true,"SCHOOL_PROFILES":false,"LEAGUE_PROFILES":false,"REC_VARSITY_COLLEGE":true,"ENABLE_SSR":"disable","SPRING_2021":true,"COLLEGE_SPRING_2021":true,"YOUTH_LEAGUES":true,"YOUTH_LEAGUE_ONBOARDING":true,"XBOXLIVE_AUTH":true,"PSN_AUTH":true,"HEARTHSTONE_DECK":false,"BILLING_V2":false}; window.version = "/static/js/bundle.2f8661ce.js" </script> <script async src="https://js.stripe.com/v3/"></script> <script src="/static/js/bundle.2f8661ce.js" defer></script>
Process finished with exit code 0

Do you know how to solve this?

Birdie0 commented on Mar 22, 2021 — Author

@blackdackota by the output i can tell that this is not server side generated page (in this case scraping would work), but client-side, so that means data comes from api requests then page gets filled.
In developer tools -> requests tab -> filter by xhr, I see couple of requests and one of them returns json with data for leaderboard.

Here's ReqBin example, there you can even get generated code for python etc. Request structure is simple:

- url: `https://api.playvs.com/graphql`
- method: `POST`
- headers:
  - `Content-Type: application/json`
- body:

```
{
    "operationName": "getPhaseResults",
    "variables": {
        "phaseId": "0f4468aa-079b-4391-adbc-1fe971f64059"
    },
    "query": "query getPhaseResults($phaseId: ID!) {\n  phase(id: $phaseId) {\n    id\n    name\n
}
```

Response by path `data -> phase -> phaseResults` contains all data you need. I recommend try to perform this request with some like Hoppscotch (online tool), Insomnia (installable) or Postman (installable) so you would see structure and figure out hot to extract it and format for sending to Discord.

blackdackota commented on Mar 23, 2021

Okay I think I understand, thanks.

blackdackota commented on Mar 24, 2021 • edited

Apparently I'm still confused.

So what is the Python code supposed to look like? Because I tried converting the code into Python using ReqBin but it didn't work how it should:

```
from requests.structures import CaseInsensitiveDict

url = "https://api.playvs.com/graphql"

headers = CaseInsensitiveDict()
headers["Content-Type"] = "application/json"

data = """
{
    "operationName": "getPhaseResults",
    "variables": {
        "phaseId": "0f4468aa-079b-4391-adbc-1fe971f64059"
    },
    "query": "query getPhaseResults($phaseId: ID!) {\n  phase(id: $phaseId) {\n    id\n    name\n
type\n    phaseResults {\n      id\n      phaseTiebreakerPoints\n      gamesWon\n      gamesLost\n
matchesWon\n      matchesLost\n      team {\n        id\n        name\n        esport {\n
id\n          slug\n          __typename\n        }\n        school {\n          id\n
name\n          logoUrl\n          slug\n          __typename\n        }\n        __typename\n
}\n      record {\n        won\n        lost\n        streak\n        winPercentage\n
matchesWonGamesWonGamesLost\n        __typename\n      }\n      __typename\n    }\n
__typename\n  }\n}\n"
}
"""


resp = requests.post(url, headers=headers, data=data)

print(resp.status_code)```

It looked to me that the main problem was that everything was a string, including the objects. So I tried to fix that but when it came to the "phaseID" and the "query" I wasn't sure what I was supposed to do with those. I want to convert the code to Python so I can use BeautifulSoup to parse it in PyCharm.

**Birdie0** commented on Mar 25, 2021 • edited

Author
```

you don't need BeautifulSoup, as data that comes from api is json and as i said before html parsing is not possible here as initial html doesn't contain data.
here's script that might fir your needs or help to reach your goal

first, install requests and tabulate packages: `pip install -U --user requests tabulate`
second, replace put url here in code with webhook url

```
import requests
from tabulate import tabulate

url = "https://api.playvs.com/graphql"
body = {
    "operationName": "getPhaseResults",
    "variables": {
        "phaseId": "0f4468aa-079b-4391-adbc-1fe971f64059"
    },
    "query": "query getPhaseResults($phaseId: ID!) {\n  phase(id: $phaseId) {\n    id\n    name\n
}
r = requests.post(url, json=body)
data = r.json()["data"]["phase"]["phaseResults"][0:10]
table_data = [[
  i + 1,
  x["team"]["name"],
  x["matchesWon"],
  x["matchesLost"],
  f"{x['gamesWon']}-{x['gamesLost']}",
  x["record"]["streak"],
  x["phaseTiebreakerPoints"]
] for i,x in enumerate(data)]
headers = ["#", "Team", "Matches Won", "Matches Lost", "Game Record", "Win Streak", "Tiebreaker Point

webhook_url = "put url here"
webhook_body = {
  "content": f"```\n{tabulate(table_data, headers=headers)}```"
}
requests.post(webhook_url, json=webhook_body)
```

▶ result:

blackdackota commented on Mar 25, 2021

Alright, thank you.

rawr37 commented on Apr 7, 2021

can you make it so that it posts at a certain time and has different content each day (I'm doing it for a daily update thing, but everyday the content is different based on a weekly schedule (webtoon))

Birdie0 commented on Apr 9, 2021 — Author

@rawr37 on ifttt with filter code you can make it choose image depending on current day from predefined list, if you want data come from somewhere outside it need you to parse website by yourself and make some script to send it with cron or similar stuff like task manager on windows.

sck1ss commented on Sep 12, 2021

> how do I ping someone using webhook message?

@Username#tag

Birdie0 commented on Sep 12, 2021 — Author

> how do I ping someone using webhook message?
>
> @Username#tag

`<@id>` reference

WindexProjects commented on Oct 2, 2021

> how do I ping someone using webhook message?

You can use `<@(Their ID)>` in the content message

RogueRaven26 commented on Oct 10, 2021

What do I put in for Event Name???

Birdie0 commented on Oct 10, 2021 — Author

@RogueRaven26 IFTTT's incoming/if Webhook has nothing to do with Discord Webhooks, which is used in action/then part.

Jossefht220 commented on Dec 4, 2021

The truth is I don't know what I'm doing haha, I don't know if the parameter is right or is everything wrong, I don't know if you can give me a hand please.

```
username = "Registro",
attachments = new[]

{
new
 {

     description = textBox1.Text,
     title = "Entrada",
     color = 1127128,
     timestamp = DateTime.Now,
     Resources = @"istockphoto-1212794677-170667a.jpg",
     playload_json = new
     {
     image = new {
     url= "Resources://istockphoto-1212794677-170667a.jpg",
      },
     }
```

```
wr.ContentType = "multipart/form-data";
wr.Method = "SET";

using (var sw = new StreamWriter(wr.GetRequestStream()))
{

   string Json = JsonConvert.SerializeObject(new
   {

     username = "Registro",
     attachments = new[]

     {
      new
       {

           description = textBox1.Text,
```

Birdie0 commented on Dec 4, 2021 Author

@Jossefht220 sorry, don't know what to fix here, what the programming language is that?

Jossefht220 commented on Dec 4, 2021

@Birdie0 c# in windows forms

Birdie0 commented on Dec 5, 2021 Author

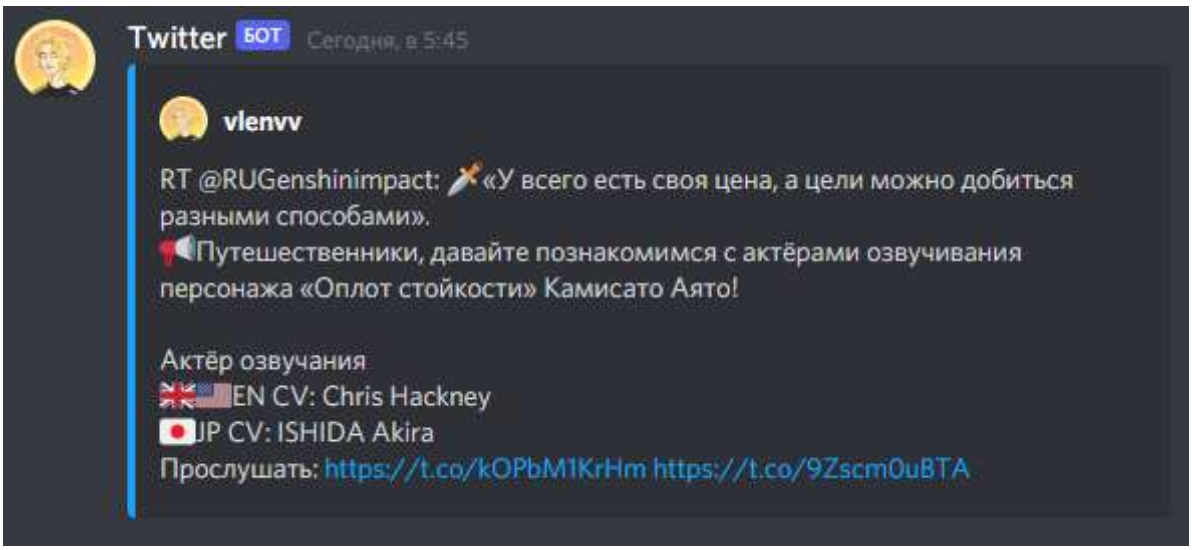
@Jossefht220 hmm, you might try checking Insomnia REST client, it has code generator, including C# for resharp and httpclient libraries. If that suits you try it.

moz9 commented on Mar 25, 2022 • edited

How do I make an image (preview tweet) or images attached to a tweet, or a video (if it's a youtube link) attached?
Here is my code:

```
{ "avatar_url": "https://avatar-resolver.vercel.app/twitter/{{UserName}}", "content":"",
"icon_url":"", "embeds": [{ "author": { "name": "{{UserName}}", "url": "{{LinkToTweet}}", "icon_url":
"https://avatar-resolver.vercel.app/twitter/{{UserName}}" }, "color": 1942002, "description":"
<<<{{Text}}>>>" }] }
```

and here is how it looks like:



Birdie0 commented on Mar 25, 2022 Author

Hi, there's no way to do that with custom embed, try using native one, which can show tweet images (including when there's more than one) and videos:

```
{
  "avatar_url": "https://avatar-resolver.vercel.app/twitter/{{UserName}}",
  "content": "{{LinkToTweet}}"
}
```

but that still won't work with external videos, the workaround for this is quite ugly, it's putting tweet content in message directly, so Discord would generate embeds for them:

```
{
  "avatar_url": "https://avatar-resolver.vercel.app/twitter/{{UserName}}",
  "content": "<<<{{Text}}>>>"
}
```

moz9 commented on Mar 25, 2022 • edited

Birdie0, thanks for the reply. I tried the first option - it just inserts a link over the body of the "embeds" array, but for some reason there is no embedding on the link, that's the problem. The feeling that the webhook does not have enough permission to embed or something restricts it from embedding attachments by link, but so only through webhooks. If you try using the IFTTT: twitter to discard scheme, then on the contrary there are too many attachments, and it looks terrible.
The second method is using

```
{
"avatar_url": "https://avatar-resolver.vercel.app/twitter/{{UserName}}",
"content": "<<<{{Text}}><{{Text}}>><{{Text}}>>><{{Text}}>>"
}
```

also did not give results

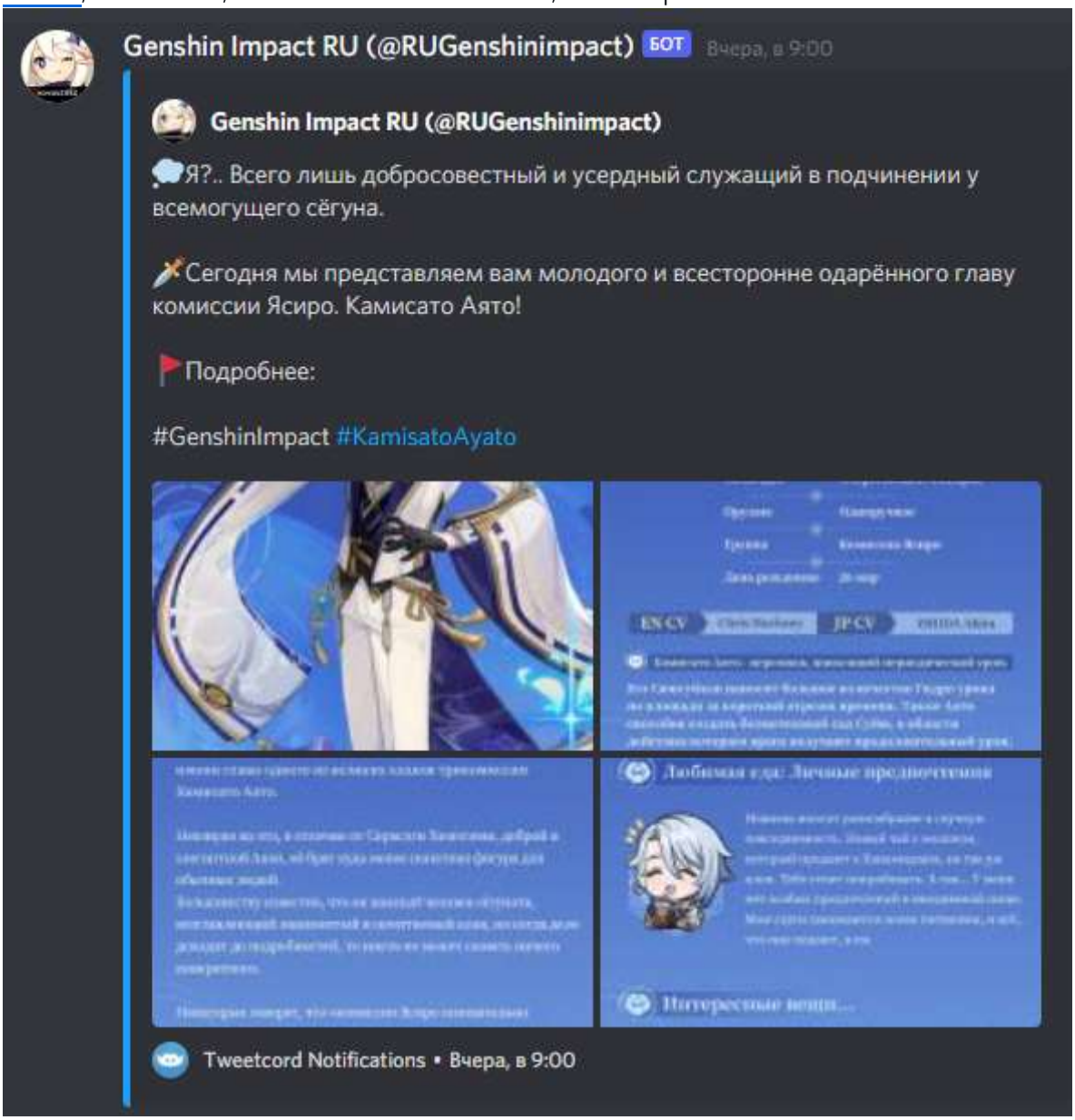
Birdie0 commented on Mar 25, 2022 Author

Using embeds block prevents generating automatic ones, also why you copied text it over and over?

Only solution that might work I have on my mind is using filter code to filter out links from tweet and then append them to content so both tweet and links in it get embed, but there's a problem that tweet images also included in text and they're shortened with t.co, so distinguish them is not possible. Appending them will result in duplicate embeds.

moz9 commented on Mar 25, 2022

Birdie0, but after all, some bots somehow do this, for example Tweetcord



Birdie0 commented on Mar 25, 2022 — Author

Tweetcord uses Twitter API directly, building rich embeds is not a problem for them, IFTTT provides preselected number of ingredients which doesn't allow to reproduce same result. Services like Pipedream or Integromat/Make give direct access to API response, that comes with flexibility also with certain difficulty of configuration if you don't know what you're doing.

moz9 commented on Mar 25, 2022

Birdie0 I see. thank you for the information and attention

floatbeta commented on Apr 20, 2022

It there a way to GET from a Discord webhook?

I created a form that gets filled out and sent to a Discord channel, now I need it to print back to the website, how can I do it?

My website is static html.

Birdie0 commented on Apr 21, 2022 Author

@floatbeta webhook is one-way message sending for such things you likely need bot to access message from channel, also storage so you don't refetch same message over on over, and something that will handle requests from page (can be server or you can use as vercel/netlify's serverless functions as alternaive), so no credentials (including webhook url is revealed). static websites is not limit for loading dynamic data, there's bunch of js frameworks allowing to organize that.

p4rkjeno commented on Jul 19, 2022

How to add fields

Birdie0 commented on Jul 19, 2022 Author

@itsEci there's `fields` block for that, could you provide some more info about what you're trying to do?