GIPHY Developers

- juanidiaz
    - Dashboard
    - Log Out

Create an App

- Docs
- FAQ
- API Explorer
- Blog

- Getting Started with the GIPHY API
- API Highlights
- GIPHY Sticker API
- GIPHY Upload API
- Code Examples
- Sample Responses
- Rendition Guide
- Language Support
- GIPHY SDKs
- Sharing and Promoting your GIPHY API Project
- Technical Documentation
- Schema Definitions

Create an App

Getting Started with the GIPHY API

Congratulations, you've made the first step toward supercharging your project or app with the power of GIPHY!

GIPHY's GIF library is the largest in the world and includes millions of original GIFs directly from the world's best content partners, original GIF artists, as well as the best GIFs from across the entire internet.

GIPHY's APIs make it dead simple for developers to incorporate this vast library right inside of their apps to deliver highly interactive content that is proven to increase daily engagement across all types of apps; messaging, chat, dating, creation, community, and more.

GIPHY Labs is a great place to see various implementations of the GIPHY API. Learn more about the rest in the documentation below.

The GIPHY API implements a REST-like interface. Connections can be made with any HTTP or HTTPS enabled programming language. The GIPHY API also implements CORS, allowing you to connect to GIPHY from JavaScript / Web browsers on your own domain. The GIPHY API provides multiple file sizes, dimensions, and formats of every GIF to meet every clients potential needs. You can view a guide to the rendition offering here.

- **Search** replicates the search found on GIPHY.
- **Translate** converts words and phrases to GIFs and is designed to be used in messaging apps, e.g. the GIPHY Slack integration.
- **Trending** pulls in the best GIFs from around the internet, hand curated by the GIPHY editorial team.
- **Random** returns a single random GIF, optionally limited to a specified tag.
- **Stickers** are a separate library of animated, transparent GIFs; great for creation apps and iOS10 iMessage apps! You can also view them on the web by visiting giphy.com/stickers.
- **Upload** allows you to upload your content programatically to GIPHY!

Other features include:

- **Emoji support** across Search, Translate, and Random endpoints!
- **Language support** for 30+ languages - see language documentation here.
- **Optimized renditions** for mobile to deliver GIFs that load fast and consume less bandwidth - see rendition guide here.
- **MP4s and Webps** of every GIF!
- **MPAA rating filters** supported across most endpoints

Access and API Keys

Get an API Key

To use the GIPHY API, you'll need to obtain an API Key by creating an app. You'll need a GIPHY account to create an app (don't worry, it's free!) Each application you create will have its own API Key. Don't worry if you're not sure what you want to use the GIPHY API for yet, all that's needed to create an app is a name and a basic description, which can be changed at any time.

Please use this key while you develop your application and experiment with your integrations. Note: your API Key is subject to rate limit constraints and we do not encourage live production deployments to use this key. If you're planning on a large-scale deployment, read on for information about obtaining a Production Key.

Request a Production Key

Once you're ready to use the GIPHY API in production, please visit your dashboard to request a production API key for your application. In your submission, please be prepared to provide the following:

- Your app name with brief description, web / app store links, etc.
- The 'live date' of the app and feature that integrates with the API. Briefly describe how the GIPHY API integrates with your app and provide screenshots of the implementation.
- As per our section 5 A of our terms of service, **we require all apps that use the GIPHY API to conspicuously display "Powered By GIPHY" attribution marks where the API is utilized.** You can find approved official logo marks here. Please provide screenshots of your attribution placement.

If you have any questions please feel free to contact us at api@giphy.com. Please submit API corrections via GitHub issues. Please see our terms of service for any restrictions on using the service. We also recommend using the JSONview plugin for Firefox or Chrome to view the API responses in your browser.

API Highlights

The GIPHY Search Endpoint is the granddaddy of them all, allowing your project or app to leverage the powerful GIF search found on giphy.com to your liking. GIPHY Search enables your users to instantly find the exact GIF they're looking for and discover new favorites. They'll never want to leave.

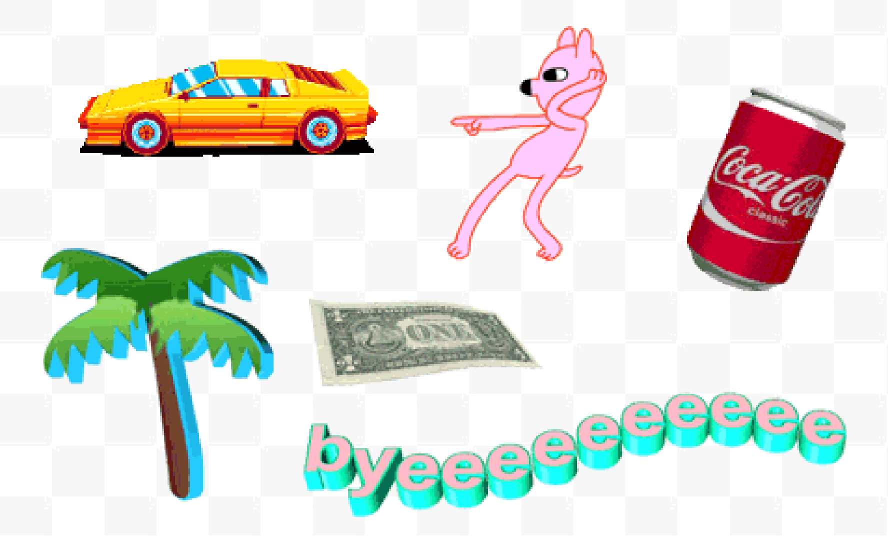
GIPHY's Trending Endpoint provides a collection of up-to-the-minute memes, tropes, and touchstones of pop culture, editorialized by real humans. Your users will have instant access to all of it.

The data returned mirrors the GIFs showcased on the GIPHY homepage. Returns 25 results by default.

Text and Emojis, step back! The GIPHY Translate Endpoint draws on search, but uses the GIPHY "special sauce" to handle translating from one vocabulary to another. In this case, words and phrases to GIFs. Example implementations of translate can be found in the GIPHY Slack, Hipchat, Wire, or Dasher integrations. Use a plus or url encode for phrases.

Need a random GIF on the fly? Say no more. The Random Endpoint returns a random GIF, limited by tag. Tell us what kind of GIF you're looking for by keyword, and we'll spin the wheel to deliver you something special and new every time. You never know what you're going to get, but we know you you're going to like it.

GIPHY Sticker API



The GIPHY Sticker API is an API that provides only animated stickers (aka animated GIFs with transparent backgrounds) in the responses. Stickers are great for messaging apps, games, and other times where a background or a square frame just won't do.

The available endpoints are as follows:

- **Sticker Search**
- **Sticker Roulette (Random)**
- **Sticker Trending**
- **Sticker Translate**

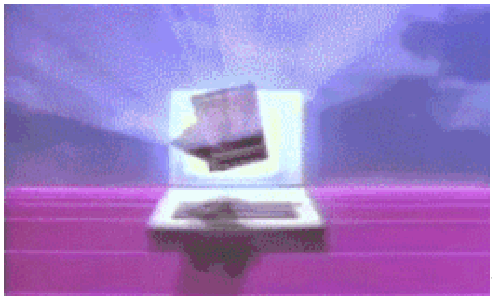
Also available are are Sticker Packs: collections of Stickers for use in your app, including the Trending and Reactions Sticker Packs, hand-curated by the GIPHY Editorial team! The endpoints relating to Sticker Packs are:

- **Sticker Packs: Listing Endpoint**
- **Sticker Packs: Individual Sticker Pack Endpoint**
- **Sticker Packs: Stickers Endpoint**
- **Sticker Packs: Children Pack Listing Endpoint**

Like the primary GIPHY API, the Sticker API is open to the public. All you need is an API Key, which you can get for free by creating an app.

Please use this key while you develop your application and experiment with your integrations. Note: the key granted to you by this developer portal is subject to rate limit constraints. If you require production-level scale for your application, apply for a production key from your dashboard. If you have more questions about rate-limits and other distinctions between API Keys, please read our FAQ.

GIPHY Upload API



"The GIPHY Upload API allows you to build on top of the world's largest and fastest GIF hosting platform...for free! Upload GIFs up to 100MB or convert videos to GIFs on the fly. Leverage our extensive rendition options to retrieve mp4s, webp, embeds, and all sorts of sizes large and small of your content. Hosted GIPHY URLs are supported and play on every major social network.

Code Examples

Below are code samples in Python, JavaScript, Ruby, PHP and the command line on connecting to the API to make a search query for "ryan gosling". Also note that we provide open-source, language-specific code in the form of our GIPHY SDKs that make connecting to and interacting with our API even simpler in your Javascript, Python, Ruby and PHP projects.

Command line, cURL

```
# curl, command line
curl "http://api.giphy.com/v1/gifs/search?q=ryan+gosling&api_key=YOUR_API_KEY&limit=5"
```

Python

(View the GIPHY Python SDK)

```
# python
import urllib,json
data = json.loads(urllib.urlopen("http://api.giphy.com/v1/gifs/search?q=ryan+gosling&api_key=YOUR_API_KEY&limit=5").read())
print json.dumps(data, sort_keys=True, indent=4)
```

Javascript

(View the GIPHY Javascript SDK)

```
//javascript, jQuery
var xhr = $.get("http://api.giphy.com/v1/gifs/search?q=ryan+gosling&api_key=YOUR_API_KEY&limit=5");
xhr.done(function(data) { console.log("success got data", data); });
```

Ruby

(View the GIPHY Ruby SDK)

```
#ruby
require 'net/http'
require 'json'
url = "http://api.giphy.com/v1/gifs/search?q=ryan+gosling&api_key=YOUR_API_KEY&limit=5"
resp = Net::HTTP.get_response(URI.parse(url))
buffer = resp.body
result = JSON.parse(buffer)
puts result
```

PHP

(View the GIPHY PHP SDK)

```
// php
$url = "http://api.giphy.com/v1/gifs/search?q=ryan+gosling&api_key=YOUR_API_KEY&limit=5";
print_r(json_decode(file_get_contents($url)));
```

Sample Responses

Below are some sample responses for some typical calls to the GIPHY API. Most responses return either a GIF Object or a list of GIF Objects.

Sample Search Endpoint Response

This call to `http://api.giphy.com/v1/gifs/search?q=funny+cat&api_key=YOUR_API_KEY` returns 25 GIF Objects for the search term `funny cat`. Each GIF Object contains an Images Object containing URLs for various versions of the GIF as described in our Rendition Guide.

```
{
    "data": [
        {
            type: "gif",
            id: "FiGiRei2ICzzG",
            slug: "funny-cat-FiGiRei2ICzzG",
            url: "http://giphy.com/gifs/funny-cat-FiGiRei2ICzzG",
            bitly_gif_url: "http://gph.is/1fIdLOl",
            bitly_url: "http://gph.is/1fIdLOl",
            embed_url: "http://giphy.com/embed/FiGiRei2ICzzG",
            username: "",
            source: "http://tumblr.com",
            rating: "g",
            caption: "",
            content_url: "",
            source_tld: "tumblr.com",
            source_post_url: "http://tumblr.com",
            import_datetime: "2014-01-18 09:14:20",
            trending_datetime: "1970-01-01 00:00:00",
            images: {
                fixed_height: {
                    url: "http://media2.giphy.com/media/FiGiRei2ICzzG/200.gif",
                    width: "568",
                    height: "200",
                    size: "460622",
                    mp4: "http://media2.giphy.com/media/FiGiRei2ICzzG/200.mp4",
                    mp4_size: "13866",
                    webp: "http://media2.giphy.com/media/FiGiRei2ICzzG/200.webp",
                    webp_size: "367786"
                },
                fixed_height_still: {
                    url: "http://media2.giphy.com/media/FiGiRei2ICzzG/200_s.gif",
                    width: "568",
                    height: "200"
                },
                fixed_height_downsampled: {
                    url: "http://media2.giphy.com/media/FiGiRei2ICzzG/200_d.gif",
                    width: "568",
                    height: "200",
                    size: "476276",
                webp: "http://media2.giphy.com/media/FiGiRei2ICzzG/200_d.webp",
                webp_size: "100890"
            },
            fixed_width: {
                url: "http://media2.giphy.com/media/FiGiRei2ICzzG/200w.gif",
                width: "200",
                height: "70",
                size: "90483",
                mp4: "http://media2.giphy.com/media/FiGiRei2ICzzG/200w.mp4",
                mp4_size: "14238",
                webp: "http://media2.giphy.com/media/FiGiRei2ICzzG/200w.webp",
                webp_size: "47302"
            },
            fixed_width_still: {
                url: "http://media2.giphy.com/media/FiGiRei2ICzzG/200w_s.gif",
                width: "200",
                height: "70"
            },
            fixed_width_downsampled: {
                url: "http://media2.giphy.com/media/FiGiRei2ICzzG/200w_d.gif",
                width: "200",
                height: "70",
                size: "71069",
                webp: "http://media2.giphy.com/media/FiGiRei2ICzzG/200w_d.webp",
                webp_size: "13186"
            },
            fixed_height_small: {
                url: "http://media2.giphy.com/media/FiGiRei2ICzzG/100.gif",
                width: "284",
                height: "100",
                size: "460622",
                webp: "http://media2.giphy.com/media/FiGiRei2ICzzG/100.webp",
                webp_size: "72748"
            },
            fixed_height_small_still: {
                url: "http://media2.giphy.com/media/FiGiRei2ICzzG/100_s.gif",
                width: "284",
                height: "100"
            },
            fixed_width_small: {
                url: "http://media2.giphy.com/media/FiGiRei2ICzzG/100w.gif",
                width: "100",
                height: "35",
                size: "90483",
                webp: "http://media2.giphy.com/media/FiGiRei2ICzzG/100w.webp",
                webp_size: "18298"
            },
            fixed_width_small_still: {
                url: "http://media2.giphy.com/media/FiGiRei2ICzzG/100w_s.gif",
                width: "100",
                height: "35"
            },
            downsized: {
                    url: "http://media2.giphy.com/media/FiGiRei2ICzzG/giphy.gif",
                    width: "500",
                    height: "176",
                    size: "426811"
                },
                downsized_still: {
                    url: "http://media2.giphy.com/media/FiGiRei2ICzzG/giphy_s.gif",
                    width: "500",
                    height: "176"
                },
                downsized_large: {
                    url: "http://media2.giphy.com/media/FiGiRei2ICzzG/giphy.gif",
                    width: "500",
                    height: "176",
                    size: "426811"
                },
                original: {
                    url: "http://media2.giphy.com/media/FiGiRei2ICzzG/giphy.gif",
                    width: "500",
                    height: "176",
                    size: "426811",
                    frames: "22",
                    mp4: "http://media2.giphy.com/media/FiGiRei2ICzzG/giphy.mp4",
                    mp4_size: "51432",
                    webp: "http://media2.giphy.com/media/FiGiRei2ICzzG/giphy.webp",
                    webp_size: "291616"
                },
                original_still: {
                    url: "http://media2.giphy.com/media/FiGiRei2ICzzG/giphy_s.gif",
                    width: "500",
                    height: "176"
                }
            },
            title: "Funny Cat GIF",
        },
        ... 24 more items
    ],
    "meta": {
        "status": 200,
        "msg": "OK"
    },
    "pagination": {
        "total_count": 1947,
        "count": 25,
        "offset": 0
    }
}
```

Sample Get GIF by ID Endpoint Response

This call to `http://api.giphy.com/v1/gifs/feqkVgjJpYtjy?api_key=YOUR_API_KEY` just returns one [GIF Object](#) relating to the GIF with id `ffeqkVgjJpYtjy`.

```
{
    "data": {
        type: "gif",
        id: "feqkVgjJpYtjy",
        slug: "eyes-shocked-bird-feqkVgjJpYtjy",
        url: "http://giphy.com/gifs/eyes-shocked-bird-feqkVgjJpYtjy",
        bitly_gif_url: "http://gph.is/XJ200y",
        bitly_url: "http://gph.is/XJ200y",
        embed_url: "http://giphy.com/embed/feqkVgjJpYtjy",
        username: "",
        source: "http://littleanimalgifs.tumblr.com/post/17994517807",
        rating: "g",
        caption: "",
        content_url: "",
        source_tld: "littleanimalgifs.tumblr.com",
        source_post_url: "http://littleanimalgifs.tumblr.com/post/17994517807",
        import_datetime: "2013-03-21 04:03:08",
        trending_datetime: "2014-11-12 06:22:52",
        images: {
            fixed_height: {
                url: "http://media0.giphy.com/media/feqkVgjJpYtjy/200.gif",
                width: "445",
                height: "200",
                size: "445432",
                mp4: "http://media0.giphy.com/media/feqkVgjJpYtjy/200.mp4",
                mp4_size: "27279",
                webp: "http://media0.giphy.com/media/feqkVgjJpYtjy/200.webp",
                webp_size: "420734"
            },
            fixed_height_still: {
                url: "http://media0.giphy.com/media/feqkVgjJpYtjy/200_s.gif",
                width: "445",
                height: "200"
            },
            fixed_height_downsampled: {
                url: "http://media0.giphy.com/media/feqkVgjJpYtjy/200_d.gif",
                width: "445",
                height: "200",
                size: "183225",
                webp: "http://media0.giphy.com/media/feqkVgjJpYtjy/200_d.webp",
                webp_size: "89516"
            },
            fixed_width: {
                url: "http://media0.giphy.com/media/feqkVgjJpYtjy/200w.gif",
                width: "200",
                height: "90",
                size: "115885",
                mp4: "http://media0.giphy.com/media/feqkVgjJpYtjy/200w.mp4",
                mp4_size: "31919",
                webp: "http://media0.giphy.com/media/feqkVgjJpYtjy/200w.webp",
                webp_size: "122600"
            },
            fixed_width_still: {
                url: "http://media0.giphy.com/media/feqkVgjJpYtjy/200w_s.gif",
                width: "200",
                height: "90"
            },
            fixed_width_downsampled: {
                url: "http://media0.giphy.com/media/feqkVgjJpYtjy/200w_d.gif",
                width: "200",
                height: "90",
                size: "83007",
                webp: "http://media0.giphy.com/media/feqkVgjJpYtjy/200w_d.webp",
                webp_size: "26460"
            },
            fixed_height_small: {
                url: "http://media0.giphy.com/media/feqkVgjJpYtjy/100.gif",
                width: "223",
                height: "100",
                size: "445432",
                webp: "http://media0.giphy.com/media/feqkVgjJpYtjy/100.webp",
                webp_size: "129604"
            },
            fixed_height_small_still: {
                url: "http://media0.giphy.com/media/feqkVgjJpYtjy/100_s.gif",
                width: "223",
                height: "100"
            },
            fixed_width_small: {
                url: "http://media0.giphy.com/media/feqkVgjJpYtjy/100w.gif",
                width: "100",
                height: "45",
                size: "115885",
                webp: "http://media0.giphy.com/media/feqkVgjJpYtjy/100w.webp",
                webp_size: "41620"
            },
            fixed_width_small_still: {
                url: "http://media0.giphy.com/media/feqkVgjJpYtjy/100w_s.gif",
                width: "100",
                height: "45"
            },
            downsized: {
                url: "http://media0.giphy.com/media/feqkVgjJpYtjy/giphy.gif",
                width: "334",
                height: "150",
                size: "511581"
            },
            downsized_still: {
                url: "http://media0.giphy.com/media/feqkVgjJpYtjy/giphy_s.gif",
                width: "334",
                height: "150"
            },
            downsized_large: {
                url: "http://media0.giphy.com/media/feqkVgjJpYtjy/giphy.gif",
                width: "334",
                height: "150",
                size: "511581"
            },
            original: {
                url: "http://media0.giphy.com/media/feqkVgjJpYtjy/giphy.gif",
                width: "334",
                height: "150",
                size: "511581",
                frames: "27",
                mp4: "http://media0.giphy.com/media/feqkVgjJpYtjy/giphy.mp4",
                mp4_size: "97841",
                webp: "http://media0.giphy.com/media/feqkVgjJpYtjy/giphy.webp",
                webp_size: "270108"
                },
                original_still: {
                url: "http://media0.giphy.com/media/feqkVgjJpYtjy/giphy_s.gif",
                width: "334",
                height: "150"
            }
        },
        title: "Shocked Eyes GIF",
    },
    "meta": {
        "status": 200,
        "msg": "OK"
    }
}
```

Rendition Guide

When rendering GIFs, you can use any of the following designations.

Please Note

When making requests that return [GIF objects](), the accept http header can be used to toggle the content type you will receive in return. By including `accept: image/*` the binary media file will be returned, otherwise some markup embedding the GIF is returned.

- **fixed_height** - Height set to 200px. Good for mobile use.
- **fixed_height_still** - Static preview image for fixed_height
- **fixed_height_downsampled** - Height set to 200px. Reduced to 6 frames to minimize file size to the lowest. Works well for unlimited scroll on mobile and as animated previews. See GIPHY.com on mobile web as an example.
- **fixed_width** - Width set to 200px. Good for mobile use.
- **fixed_width_still** - Static preview image for fixed_width
- **fixed_width_downsampled** - Width set to 200px. Reduced to 6 frames. Works well for unlimited scroll on mobile and as animated previews.

- **fixed_height_small** - Height set to 100px. Good for mobile keyboards.
- **fixed_height_small_still** - Static preview image for fixed_height_small
- **fixed_width_small** - Width set to 100px. Good for mobile keyboards
- **fixed_width_small_still** - Static preview image for fixed_width_small
- **preview** - File size under 50kb. Duration may be truncated to meet file size requirements. Good for thumbnails and previews.
- **downsized_small** - File size under 200kb.
- **downsized** - File size under 2mb.
- **downsized_medium** - File size under 5mb.
- **downsized_large** - File size under 8mb.
- **downsized_still** - Static preview image for downsized
- **original** - Original file size and file dimensions. Good for desktop use.
- **original_still** - Preview image for original
- **looping** - Duration set to loop for 15 seconds. Only recommended for this exact use case.

Language Support

- lang - specify default country for regional content; format is 2-letter ISO 639-1 language code

Supported Languages

- English (**en**)
- Spanish (**es**)
- Portuguese (**pt**)
- Indonesian (**id**)
- French (**fr**)
- Arabic (**ar**)
- Turkish (**tr**)
- Thai (**th**)
- Vietnamese (**vi**)
- German (**de**)
- Italian (**it**)
- Japanese (**ja**)
- Chinese Simplified (**zh-CN**)
- Chinese Traditional (**zh-TW**)
- Russian (**ru**)
- Korean (**ko**)
- Polish (**pl**)
- Dutch (**nl**)
- Romanian (**ro**)

- Hungarian (**hu**)
- Swedish (**sv**)
- Czech (**cs**)
- Hindi (**hi**)
- Bengali (**bn**)
- Danish (**da**)
- Farsi (**fa**)
- Filipino (**tl**)
- Finnish (**fi**)
- Hebrew (**iw**)
- Malay (**ms**)
- Norwegian (**no**)
- Ukrainian (**uk**)

GIPHY SDKs

GIPHY also offers several open-source platform-specific SDKs that make integrating the GIPHY API even easier. Available for iOS, Android and various other programming languages, the code can be found in the following Github repositories under the Mozilla Public License, v. 2.0. Where applicable, binaries are published to public repositories.

- **iOS**: https://github.com/Giphy/giphy-ios-sdk-core
- **Android**: https://github.com/Giphy/giphy-android-sdk-core
- **Javascript**: https://github.com/Giphy/giphy-js-sdk-core
- **Python**: https://github.com/Giphy/giphy-python-client
- **Ruby**: https://github.com/Giphy/giphy-ruby-client
- **PHP**: https://github.com/Giphy/giphy-php-client

With the GIPHY SDKs, there's no need to re-invent the wheel: if you use GIPHY in your application, the SDK provides you with a library that will have you implementing features faster. The SDKs are all open-source, so if you think a feature is missing or have an idea for an improvement, feel free to make a pull request and send it our way!

Sharing and Promoting your GIPHY API Project

Have you built something awesome with the GIPHY API? Get in touch so your project can join the hallowed ranks of GIPHY Labs, our testing ground and showcase for all things GIPHY!

Technical Documentation / Specifications

Here you'll find the nitty gritty of the GIPHY API's various endpoints, including required parameters and explanations of the responses you'll receive.

Summary

Path	Operation	Description
/gifs/search	GET	Search Endpoint

Path	Operation	Description
/gifs/trending	GET	Trending GIFs Endpoint
/gifs/translate	GET	Translate Endpoint
/gifs/random	GET	Random Endpoint
/gifs/{gif_id}	GET	Get GIF by ID Endpoint
/gifs	GET	Get GIFs by ID Endpoint
/stickers/search	GET	Sticker Search Endpoint
/stickers/trending	GET	Trending Stickers Endpoint
/stickers/translate	GET	Sticker Translate Endpoint
/stickers/random	GET	Random Sticker Endpoint
upload.giphy.com/v1/gifs/	POST	Upload API: Upload Endpoint

Search Endpoint

Host

api.giphy.com

Path

GET /v1/gifs/search

Description

Search all GIPHY GIFs for a word or phrase. Punctuation will be stripped and ignored. Use a plus or url encode for phrases. Example paul+rudd, ryan+gosling or american+psycho.

	Example
api_key: string GIPHY API Key.	`YOUR_API_KEY`
q: string Search query term or phrase. GIPHY search will automatically look for exact matches to queries + AND match + OR match. Explicit AND + OR boolean clauses in search queries are not supported.	`cheeseburgers`
limit: integer (int32) The maximum number of records to return. 25	`20`
offset: integer (int32) An optional results offset. Defaults to 0.	`5`
rating: string Filters results by specified rating.	`g`
lang: string Specify default language for regional content; use a 2-letter ISO 639-1 language code. See list of supported languages here.	`en`
fmt: string Used to indicate the expected response format. Default is Json.	`json`

data: object[]

- GIF Object
- pagination: Pagination Object
- meta: Meta Object

Trending Endpoint

Host

api.giphy.com

Path

GET /v1/gifs/trending

Description

Fetch GIFs currently trending online. Hand curated by the GIPHY editorial team. The data returned mirrors the GIFs showcased on the GIPHY homepage. Returns 25 results by default.

	Example
api_key: string GIPHY API Key.	`YOUR_API_KEY`
limit: integer (int32) The maximum number of records to return. 25	`20`
offset: integer (int32) An optional results offset. 0	`50`
rating: string Filters results by specified rating.	`g`

	Example
fmt: string Used to indicate the expected response format. Default is Json.	`json`

data: object[]

- [GIF Object]
- pagination: [Pagination Object]
- meta: [Meta Object]

Translate Endpoint

Host

api.giphy.com

Path

GET /v1/gifs/translate

Description

The translate API draws on search, but uses the GIPHY `special sauce` to handle translating from one vocabulary to another. In this case, words and phrases to GIFs.

	Example
api_key: string GIPHY API Key.	`YOUR_API_KEY`
s: string Search term.	`ryan gosling`
weirdness: int value from 0-10 which makes results more or less weird/random/wtf	`10`

data: object[]

- [GIF Object]
- meta: [Meta Object]

Random Endpoint

Host

api.giphy.com

Path

GET /v1/gifs/random

Description

Returns a random GIF, limited by tag. Excluding the tag parameter will return a random GIF from the GIPHY catalog.

	Example
api_key: string GIPHY API Key.	`YOUR_API_KEY`
tag: string Filters results by specified tag.	`burrito`
rating: string Filters results by specified rating.	`g`
fmt: string Used to indicate the expected response format. Default is Json.	`json`

data: object[]

- [GIF Object]

- meta: Meta Object

Get GIF by ID Endpoint

Host

api.giphy.com

Path

GET /v1/gifs/{gif_id}

Description

Returns a GIF given that GIF's unique ID

	Example
api_key: string GIPHY API Key.	`YOUR_API_KEY`
gif_id: string Filters results by specified GIF ID.	`xT4uQulxzV39haRFjG`

data: object[]

- GIF Object
- meta: Meta Object

Get GIFs by ID Endpoint

Host

api.giphy.com

Path

GET /v1/gifs

Description

A multiget version of the get GIF by ID endpoint.

	Example
api_key: string GIPHY API Key.	`YOUR_API_KEY`
ids: string Filters results by specified GIF IDs, separated by commas.	`xT4uQulxzV39haRFjG,3og0IPxMM0erATueVW`

data: object[]

- GIF Object
- pagination: Pagination Object
- meta: Meta Object

Stickers: Search Endpoint

Host

api.giphy.com

Path

GET /v1/stickers/search

Description

Replicates the functionality and requirements of the classic GIPHY search, but returns animated stickers rather than GIFs.

Example

	Example
api_key: string GIPHY API Key.	`YOUR_API_KEY`
q: string Search query term or prhase.	`cheeseburgers`
limit: integer (int32) The maximum number of records to return. 25	`20`
offset: integer (int32) An optional results offset. Defaults to 0.	`5`
rating: string Filters results by specified rating.	`g`
lang: string Specify default country for regional content; use a 2-letter ISO 639-1 country code. See list of supported languages here.	`en`
fmt: string Used to indicate the expected response format. Default is Json.	`json`

data: object[]

- GIF Object
- pagination: Pagination Object
- meta: Meta Object

Stickers: Trending Endpoint

Host

api.giphy.com

Path

GET /v1/stickers/trending

Description

Fetch Stickers currently trending online. Hand curated by the GIPHY editorial team. Returns 25 results by default.

	Example
api_key: string GIPHY API Key.	`YOUR_API_KEY`
limit: integer (int32) The maximum number of records to return. 25	`20`
rating: string Filters results by specified rating.	`g`
fmt: string Used to indicate the expected response format. Default is Json.	`json`

data: object[]

- GIF Object
- meta: Meta Object

Stickers: Translate Endpoint

Host

api.giphy.com

Path

GET /v1/stickers/translate

Description

The translate API draws on search, but uses the GIPHY *special sauce* to handle translating from one vocabulary to another. In this case, words and phrases to GIFs.

	Example
api_key: string GIPHY API Key.	`YOUR_API_KEY`
s: string Search term.	`ryan gosling`

data: object[]

- [GIF Object]
- meta: [Meta Object]

Stickers: Random Endpoint

Host

api.giphy.com

Path

GET /v1/stickers/random

Description

Returns a random Sticker, limited by tag. Excluding the tag parameter will return a random Sticker from the GIPHY catalog.

Example

	Example
api_key: string GIPHY API Key.	`YOUR_API_KEY`
tag: string Filters results by specified tag.	`burrito`
rating: string Filters results by specified rating.	`g`
fmt: string Used to indicate the expected response format. Default is Json.	`json`

data: object[]

- GIF Object
- meta: Meta Object

Sticker Packs: Listing Endpoint

Host

api.giphy.com

Path

GET /v1/stickers/packs

Description

Returns a list of all sticker packs available. Hand-curated by the GIPHY editorial team.

Example

	Example
api_key: string GIPHY API Key.	`YOUR_API_KEY`

data: object[]

- Child Pack Objects
- meta: Meta Object

Sticker Packs: Individual Sticker Pack Endpoint

Host

api.giphy.com

Path

GET /v1/stickers/packs/{pack_id}

Description

Returns the metadata for any Sticker pack. Note that convenience URLs for GIPHY's curated Trending and Reactions Sticker Packs exist at `/v1/stickers/packs/trending` and `/v1/stickers/packs/reactions`, respectively.

	Example
api_key: string GIPHY API Key.	`YOUR_API_KEY`
pack_id: integer (int32) Filters results by specified Sticker Pack ID.	`3153`
limit: integer (int32) The maximum number of records to return.	`20`

25

	Example
offset: integer (int32) An optional results offset. Defaults to 0.	`5`

data: object[]

- [Sticker Pack Metadata Objects]
- meta: [Meta Object]

Sticker Packs: Stickers Endpoint

Host

api.giphy.com

Path

GET /v1/stickers/packs/{pack_id}/stickers

Description

Returns the stickers within an individual sticker pack. Note that convenience URLs for GIPHY's curated Trending and Reactions Sticker Packs exist at `/v1/stickers/packs/trending/stickers` and `/v1/stickers/packs/reactions/stickers`, respectively.

	Example
api_key: string GIPHY API Key.	`YOUR_API_KEY`
pack_id: integer (int32) Filters results by specified Sticker Pack ID.	`3153`
limit: integer (int32) The maximum number of records to return.	`20`
25	

	Example
offset: integer (int32) An optional results offset. Defaults to 0.	`5`

data: object[]

- [GIF Objects]
- pagination: [Pagination Object]
- meta: [Meta Object]

Sticker Packs: Children Pack Listing Endpoint

Host

api.giphy.com

Path

GET /v1/stickers/packs/{pack_id}/children

Description

A Sticker pack is a recursive data structure and so packs may contain other packs. For example, the 'Reactions' pack would have an 'OMG' child pack. This endpoint lists all children packs of a given Sticker pack. Note that convenience URLs for GIPHY's curated Trending and Reactions Sticker Packs exist at `/v1/stickers/packs/trending/children` and `/v1/stickers/packs/reactions/children`, respectively.

	Example
api_key: string GIPHY API Key.	`YOUR_API_KEY`
pack_id: integer (int32) Filters results by specified Sticker Pack ID.	`3153`

data: object[]

- Child Pack Objects
- meta: Meta Object

Upload API: Upload Endpoint

Host

api.giphy.com

Path

POST upload.giphy.com/v1/gifs

Description

The GIPHY Upload Endpoint allows you to upload and host your content programmatically to giphy.com. We accept animated gifs or video files up to 100MB. **Note:** If you're using a rate-limited key assigned by the developer portal, you will not be able to specify a GIPHY channel username to your request and you will be limited to 10 uploads per day. Apply for a production key from your dashboard to have these limits lifted. Only approved apps will need to include a GIPHY channel username. You can use the endpoint to upload your content, attach tags and other meta tag in a single HTTP or HTTPS POST request.

	Example
api_key: string GIPHY API Key.	`YOUR_API_KEY`
username: string Your assigned username (required for approved apps only).	`JoeCool3000`
file: string (binary) The animated GIF or video file you'd like to upload. (Local file resource, required if no source_image_url supplied).	
source_image_url: string The URL for the image or video you wish to upload (required if no file parameter specified)	`http://www.mysite.com/myfile.mp4`

	Example
tags: string A comma delimited list of tags to be applied to the upload	`pets, cat, meow`
source_post_url: string The URL source of the asset.	`http://www.mysite.com/my-blog-post/`

Meta Object

Response Codes

The API will return a Meta Object in its response containing one of the following codes to describe the result of your request.

Response Code	Type	Description
200	OK	Your request was successful.
400	Bad Request	Your request was formatted incorrectly or missing required parameters.
403	Forbidden	You weren't authorized to make your request; most likely this indicates an issue with your API Key.
404	Not Found	The particular GIF you are requesting was not found. This occurs, for example, if you request a GIF by an id that does not exist.
429	Too Many Requests	Your API Key is making too many requests. Read about requesting a Production Key to upgrade your API Key rate limits.

Schema definitions

The GIF Object

The GIF Object is returned from most of the GIPHY API Endpoints and contains a variety of information, including the images object, which contains URLs for GIFs in many different formats and sizes.

Property
Example
type: string gif

By default, this is almost always gif

```
"gif"
```

id: string

This GIF's unique ID

`"YsTs5ltWtEhnq"`

slug: string

The unique slug used in this GIF's URL

`"confused-flying-YsTs5ltWtEhnq"`

url: string

The unique URL for this GIF

`"http://giphy.com/gifs/confused-flying-YsTs5ltWtEhnq"`

bitly_url: string

The unique bit.ly URL for this GIF

`"http://gph.is/1gsWDcL"`

embed_url: string

A URL used for embedding this GIF

`"http://giphy.com/embed/YsTs5ltWtEhnq"`

username: string

The username this GIF is attached to, if applicable

`"JoeCool4000"`

source: string

The page on which this GIF was found

`"http://www.reddit.com/r/reactiongifs/comments/1xpyaa/superman_goes_to_hollywood/"`

rating: string

The MPAA-style rating for this content. Examples include Y, G, PG, PG-13 and R

```
"g"
```

content_url: string

Currently unused

user: User

An object containing data about the user associated with this GIF, if applicable.

source_tld: string

The top level domain of the source URL.

```
"cheezburger.com"
```

source_post_url: string

The URL of the webpage on which this GIF was found.

```
"http://cheezburger.com/5282328320"
```

update_datetime: string

The date on which this GIF was last updated.

```
"2013-08-01 12:41:48"
```

create_datetime: string

The date this GIF was added to the GIPHY database.

```
"2013-08-01 12:41:48"
```

import_datetime: string

The creation or upload date from this GIF's source.

```
"2013-08-01 12:41:48"
```

trending_datetime: string

The date on which this gif was marked trending, if applicable.

```
"2013-08-01 12:41:48"
```

images: Images

An object containing data for various available formats and sizes of this GIF.

title: string

The title that appears on giphy.com for this GIF.

```
"Happy Dancing GIF"
```

The Images Object

The Images Object is included in the GIF Object and contains a series of objects that contain URLs and other information for GIFs in many different formats and sizes as described in our rendition guide.

Object
Property
Example
fixed_height: object

Data surrounding versions of this GIF with a fixed height of 200 pixels. Good for mobile use.

url: string

The publicly-accessible direct URL for this GIF.

```
"https://media1.giphy.com/media/cZ7rmKfFYOvYI/200.gif"
```

width: string

The width of this GIF in pixels.

`"320"`

height: string

The height of this GIF in pixels.

`"200"`

size: string

The size of this GIF in bytes.

`"32381"`

mp4: string

The URL for this GIF in .MP4 format.

`"https://media1.giphy.com/media/cZ7rmKfFYOvYI/200.mp4"`

mp4_size: string

The size in bytes of the .MP4 file corresponding to this GIF.

`"25123"`

webp: string

The URL for this GIF in .webp format.

`"https://media1.giphy.com/media/cZ7rmKfFYOvYI/200.webp"`

webp_size: string

The size in bytes of the .webp file corresponding to this GIF.

`"12321"`

fixed_height_still: object

Data surrounding a static image of this GIF with a fixed height of 200 pixels.

url: string

The publicly-accessible direct URL for this GIF.

```
"https://media1.giphy.com/media/cZ7rmKfFYOvYI/200_s.gif"
```

width: string

The width of this GIF in pixels.

```
"320"
```

height: string

The height of this GIF in pixels.

```
"200"
```

fixed_height_downsampled: object

Data surrounding versions of this GIF with a fixed height of 200 pixels and the number of frames reduced to 6.

url: string

The publicly-accessible direct URL for this GIF.

```
"https://media1.giphy.com/media/cZ7rmKfFYOvYI/200_d.gif"
```

width: string

The width of this GIF in pixels.

```
"320"
```

height: string

The height of this GIF in pixels.

```
"200"
```

size: string

The size of this GIF in bytes.

```
"32381"
```

webp: string

The URL for this GIF in .webp format.

```
"https://media1.giphy.com/media/cZ7rmKfFYOvYI/200_d.webp"
```

webp_size: string

The size in bytes of the .webp file corresponding to this GIF.

```
"12321"
```

fixed_width: object

Data surrounding versions of this GIF with a fixed width of 200 pixels. Good for mobile use.

url: string

The publicly-accessible direct URL for this GIF.

```
"https://media1.giphy.com/media/cZ7rmKfFYOvYI/200w.gif"
```

width: string

The width of this GIF in pixels.

```
"320"
```

height: string

The height of this GIF in pixels.

```
"200"
```

size: string

The size of this GIF in bytes.

```
"32381"
```

mp4: string

The URL for this GIF in .MP4 format.

```
"https://media1.giphy.com/media/cZ7rmKfFYOvYI/200w.mp4"
```

mp4_size: string

The size in bytes of the .MP4 file corresponding to this GIF.

```
"25123"
```

webp: string

The URL for this GIF in .webp format.

```
"https://media1.giphy.com/media/cZ7rmKfFYOvYI/200w.webp"
```

webp_size: string

The size in bytes of the .webp file corresponding to this GIF.

```
"12321"
```

fixed_width_still: object

Data surrounding a static image of this GIF with a fixed width of 200 pixels.

url: string

The publicly-accessible direct URL for this GIF.

```
"https://media1.giphy.com/media/cZ7rmKfFYOvYI/200w_s.gif"
```

width: string

The width of this GIF in pixels.

```
"320"
```

height: string

The height of this GIF in pixels.

```
"200"
```

fixed_width_downsampled: object

Data surrounding versions of this GIF with a fixed width of 200 pixels and the number of frames reduced to 6.

url: string

The publicly-accessible direct URL for this GIF.

```
"https://media1.giphy.com/media/cZ7rmKfFYOvYI/200w_d.gif"
```

width: string

The width of this GIF in pixels.

```
"320"
```

height: string

The height of this GIF in pixels.

```
"200"
```

size: string

The size of this GIF in bytes.

```
"32381"
```

webp: string

The URL for this GIF in .webp format.

```
"https://media1.giphy.com/media/cZ7rmKfFYOvYI/200w_d.webp"
```

webp_size: string

The size in bytes of the .webp file corresponding to this GIF.

```
"12321"
```

fixed_height_small: object

Data surrounding versions of this GIF with a fixed height of 100 pixels. Good for mobile keyboards.

url: string

The publicly-accessible direct URL for this GIF.

```
"https://media1.giphy.com/media/cZ7rmKfFYOvYI/100.gif"
```

width: string

The width of this GIF in pixels.

```
"320"
```

height: string

The height of this GIF in pixels.

```
"100"
```

size: string

The size of this GIF in bytes.

```
"32381"
```

mp4: string

The URL for this GIF in .MP4 format.

```
"https://media1.giphy.com/media/cZ7rmKfFYOvYI/100.mp4"
```

mp4_size: string

The size in bytes of the .MP4 file corresponding to this GIF.

```
"25123"
```

webp: string

The URL for this GIF in .webp format.

```
"https://media1.giphy.com/media/cZ7rmKfFYOvYI/100.webp"
```

webp_size: string

The size in bytes of the .webp file corresponding to this GIF.

```
"12321"
```

fixed_height_small_still: object

Data surrounding a static image of this GIF with a fixed height of 100 pixels.

url: string

The publicly-accessible direct URL for this GIF.

```
"https://media1.giphy.com/media/cZ7rmKfFYOvYI/100_s.gif"
```

width: string

The width of this GIF in pixels.

```
"320"
```

height: string

The height of this GIF in pixels.

```
"100"
```

fixed_width_small: object

Data surrounding versions of this GIF with a fixed width of 100 pixels. Good for mobile keyboards.

url: string

The publicly-accessible direct URL for this GIF.

```
"https://media1.giphy.com/media/cZ7rmKfFYOvYI/100w.gif"
```

width: string

The width of this GIF in pixels.

```
"100"
```

height: string

The height of this GIF in pixels.

```
"200"
```

size: string

The size of this GIF in bytes.

```
"32381"
```

mp4: string

The URL for this GIF in .MP4 format.

```
"https://media1.giphy.com/media/cZ7rmKfFYOvYI/100w.mp4"
```

mp4_size: string

The size in bytes of the .MP4 file corresponding to this GIF.

```
"25123"
```

webp: string

The URL for this GIF in .webp format.

```
"https://media1.giphy.com/media/cZ7rmKfFYOvYI/100w.webp"
```

webp_size: string

The size in bytes of the .webp file corresponding to this GIF.

```
"12321"
```

fixed_width_small_still: object

Data surrounding a static image of this GIF with a fixed width of 100 pixels.

url: string

The publicly-accessible direct URL for this GIF.

```
"https://media1.giphy.com/media/cZ7rmKfFYOvYI/100w_s.gif"
```

width: string

The width of this GIF in pixels.

```
"100"
```

height: string

The height of this GIF in pixels.

```
"200"
```

downsized: object

Data surrounding a version of this GIF downsized to be under 2mb.

url: string

The publicly-accessible direct URL for this GIF.

```
"https://media1.giphy.com/media/cZ7rmKfFYOvYI/giphy-downsized.gif"
```

width: string

The width of this GIF in pixels.

```
"320"
```

height: string

The height of this GIF in pixels.

```
"200"
```

size: string

The size of this GIF in bytes.

```
"12381"
```

downsized_still: object

Data surrounding a static preview image of the downsized version of this GIF.

url: string

The publicly-accessible direct URL for this GIF.

```
"https://media1.giphy.com/media/cZ7rmKfFYOvYI/giphy-downsized_s.gif"
```

width: string

The width of this GIF in pixels.

```
"320"
```

height: string

The height of this GIF in pixels.

```
"200"
```

downsized_large: object

Data surrounding a version of this GIF downsized to be under 8mb.

url: string

The publicly-accessible direct URL for this GIF.

`"https://media1.giphy.com/media/cZ7rmKfFYOvYI/giphy-downsized-large.gif"`

width: string

The width of this GIF in pixels.

`"320"`

height: string

The height of this GIF in pixels.

`"200"`

size: string

The size of this GIF in bytes.

`"32381"`

downsized_medium: object

Data surrounding a version of this GIF downsized to be under 5mb.

url: string

The publicly-accessible direct URL for this GIF.

`"https://media1.giphy.com/media/cZ7rmKfFYOvYI/giphy-downsized-medium.gif"`

width: string

The width of this GIF in pixels.

`"320"`

height: string

The height of this GIF in pixels.

`"200"`

size: string

The size of this GIF in bytes.

```
"32381"
```

downsized_small: object

Data surrounding a version of this GIF downsized to be under 200kb.

url: string

The publicly-accessible direct URL for this GIF.

```
"https://media1.giphy.com/media/cZ7rmKfFYOvYI/downsized_small.gif"
```

width: string

The width of this GIF in pixels.

```
"320"
```

height: string

The height of this GIF in pixels.

```
"200"
```

size: string

The size of this GIF in bytes.

```
"122381"
```

original: object

Data surrounding the original version of this GIF. Good for desktop use.

url: string

The publicly-accessible direct URL for this GIF.

```
"https://media1.giphy.com/media/cZ7rmKfFYOvYI/giphy.gif"
```

width: string

The width of this GIF in pixels.

```
"320"
```

height: string

The height of this GIF in pixels.

```
"200"
```

size: string

The size of this GIF in bytes.

```
"32381"
```

frames: string

The number of frames in this GIF.

```
"15"
```

mp4: string

The URL for this GIF in .MP4 format.

```
"https://media1.giphy.com/media/cZ7rmKfFYOvYI/giphy.mp4"
```

mp4_size: string

The size in bytes of the .MP4 file corresponding to this GIF.

```
"25123"
```

webp: string

The URL for this GIF in .webp format.

```
"https://media1.giphy.com/media/cZ7rmKfFYOvYI/giphy.webp"
```

webp_size: string

The size in bytes of the .webp file corresponding to this GIF.

```
"12321"
```

original_still: object

Data surrounding a static preview image of the original GIF.

url: string

The publicly-accessible direct URL for this GIF.

```
"https://media1.giphy.com/media/cZ7rmKfFYOvYI/giphy_s.gif"
```

width: string

The width of this GIF in pixels.

```
"320"
```

height: string

The height of this GIF in pixels.

```
"200"
```

looping: object

Data surrounding a version of this GIF set to loop for 15 seconds.

mp4: string

The URL for this GIF in .MP4 format.

```
"https://media1.giphy.com/media/cZ7rmKfFYOvYI/looping.mp4"
```

preview: object

Data surrounding a version of this GIF in .MP4 format limited to 50kb that displays the first 1-2 seconds of the GIF.

mp4: string

The URL for this GIF in .MP4 format.

```
"https://media1.giphy.com/media/cZ7rmKfFYOvYI/giphy-preview.mp4"
```

mp4_size: string

The size of this file in bytes.

```
"20241"
```

width: string

The width of this file in pixels.

```
"320"
```

height: string

The height of this file in pixels.

```
"200"
```

preview_gif: object

Data surrounding a version of this GIF limited to 50kb that displays the first 1-2 seconds of the GIF.

url: string

The URL for this preview GIF.

```
"https://media1.giphy.com/media/cZ7rmKfFYOvYI/giphy-preview.gif
```

width: string

The width of this GIF in pixels.

```
"320"
```

height: string

The height of this GIF in pixels.

```
"200"
```

size: string

The size of this file in bytes.

```
"10291"
```

The Meta Object

The Meta Object contains basic information regarding the request, whether it was successful, and the response given by the API. Click here to see a description of types of response codes the API might give you under different circumstances.

Property
Example
msg: string

HTTP Response Message

```
"OK"
```

status: integer (int32)

HTTP Response Code

```
200
```

response_id: string

A unique ID paired with this response from the API.

```
"57eea03c72381f86e05c35d2"
```

The Pagination Object

The Pagination Object contains information relating to the number of total results available as well as the number of results fetched and their relative positions.

Property
Example
offset: integer (int32)

Position in pagination.

`2591`

total_count: integer (int32)

Total number of items available.

`250`

count: integer (int32)

Total number of items returned.

`25`

The User Object

The User Object contains information about the user associated with a GIF and URLs to assets such as that user's avatar image, profile, and more.

Property
Example
avatar_url: string

The URL for this user's avatar image.

`"https://media1.giphy.com/avatars/election2016/XwYrZi5H87o6.gif"`

banner_url: string

The URL for the banner image that appears atop this user's profile page.

`"https://media4.giphy.com/avatars/cheezburger/XkuejOhoGLE6.jpg"`

profile_url: string

The URL for this user's profile.

```
"https://giphy.com/cheezburger/"
```

username: string

The username associated with this user.

```
"joecool4000"
```

display_name: string

The display name associated with this user (contains formatting the base username might not).

```
"JoeCool4000"
```

twitter: string

The Twitter username associated with this user, if applicable.

```
"@joecool4000"
```

The Child Pack Object

A Sticker Pack is a collection of Stickers. Both the Sticker Pack Metadata Object and Child Pack Objects are recursive data structures that contains metadata about a Sticker Pack itself, the stickers contained therein, and potentially sub-packs. (Sticker packs may be nested arbitrarily and are navigated like any recursive data structure.) The Child Pack Object is a condensed collection of information about a Sticker Pack that appears in listings of Sticker Packs.

Property
Example
id: integer (int32)

This Sticker Pack's unique numeric ID.

```
3151
```

parent: integer (int32)

Numeric identifier for the parent Sticker Pack.

```
3150
```

type: string

Will return "community" or "editorial". This describes whether this Sticker Pack is curated by GIPHY or by the community.

```
"community"
```

content_type: string

Will always return "sticker" in this case.

```
"sticker"
```

slug: string

URL-friendly name for this Sticker Pack.

```
"love"
```

display_name: string

Human-readable name for this Sticker Pack. (May contain formatting the other names don't).

```
"Love"
```

short_display_name: string

A more concise version of this Sticker Pack's display name.

```
"Love"
```

description: string

Long form description of this Sticker Pack.

```
"The most lovely stickers on the interweb!"
```

banner_image: string

Will return a banner image for this Sticker Pack (either JPG, PNG, or GIF) with 1040x160 dimensions.

```
"https://media.giphy.com/image.jpg"
```

has_children: boolean

Describes whether or not this Sticker Pack contains child Sticker Packs.

```
True
```

user: User

The GIPHY user associated with this Sticker Pack.

featured_gif: GIF Object

The GIF that will appear in a thumbnail, header image or other visual representation when referencing this Sticker Pack.

The Sticker Pack Metadata Object

A Sticker Pack is a collection of Stickers. Both the Sticker Pack Metadata Object and Child Pack Objects are recursive data structures that contains metadata about a Sticker Pack itself, the stickers contained therein, and potentially sub-packs. (Sticker packs may be nested arbitrarily and are navigated like any recursive data structure.) The Sticker Pack Metadata Object is an expanded collection of information about a Sticker Pack.

Property
Example
id: integer (int32)

This Sticker Pack's unique numeric ID.

```
3151
```

display_name: string

Human-readable name for this Sticker Pack. (May contain formatting the other names don't).

```
"Trending"
```

slug: string

URL-friendly slug for this Sticker Pack.

```
"trending"
```

content_type: string

Will always return "sticker" in this case.

```
"sticker"
```

short_display_name: string

A more concise version of this Sticker Pack's display name.

```
"Love"
```

description: string

Long form description of this Sticker Pack.

```
"The most lovely stickers on the interweb!"
```

banner_image: string

Will return a banner image for this Sticker Pack (either JPG, PNG, or GIF) with 1040x160 dimensions.

```
"https://media.giphy.com/image.jpg"
```

has_children: boolean

Describes whether or not this Sticker Pack contains child Sticker Packs.

```
True
```

user: User

The GIPHY user associated with this Sticker Pack.

featured_gif: GIF Object

The GIF that will appear in a thumbnail, header image or other visual representation when referencing this Sticker Pack.

tags: object[]

An array of tags which make the sticker packs discoverable via a search

tag: string

The name of an individual tag associated with this Sticker Pack.

```
"weird sticker"
```

rank: integer (int32)

An ordering of the relevance of this tag to the Sticker Pack. (Begins with 0).

```
0
```

ancestors: object[]

An array of all ancestor Sticker Packs. (Sticker packs are hierarchical, all sticker parents descend from the root sticker pack "Stickers".)

id: integer (int32)

This ancestor Sticker Pack's unique numeric ID.

```
2815
```

slug: string

URL-friendly name for this ancestor Sticker Pack.

```
"stickers"
```

display_name: string

Human-readable name for this ancestor Sticker Pack. (May contain formatting the other names don't).

```
"Stickers"
```

short_display_name: string

A more concise version of this ancestor Sticker Pack's display name.

```
"Stickers"
```

featured_gif_id: string

The ID of the GIF that will appear in a thumbnail, header image or other visual representation when referencing this ancestor Sticker Pack.

```
"l0MYSiQPbNeLyZJZK"
```

parent: string

Numeric identifier for the parent Sticker Pack, if applicable.

```
2362
```

has_children: boolean

Describes whether or not this ancestor Sticker Pack contains child Sticker Packs.

```
True
```

banner_image: string

Will return a banner image for this Sticker Pack (either JPG, PNG, or GIF) with 1040x160 dimensions.

```
"https://media.giphy.com/image.jpg"
```