

Integrating Giphy with Teamchat

Step-1: Access and API Keys

The Giphy API is open to the public. API key used in the project: dc6zaTOxFJmzC. This API key is required for all endpoints. This given API key is a public beta key.

Note:- In this project public API key is used and is in working condition so you don't have to change it in the project unless you want to do the integration with a production key.

If in case this API key is not working or you want to use a production key i.e a key only for your product "Teamchat" then you have to perform certain steps:

Visit api.giphy.com/submit to request a production API key. 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 want to change the current public key defined in the project with your production key then follow the steps given below:-

- You just have to make changes in the "giphyintegrator.java" file given in the package "com.teamchat.integration.giphy.bot"
- Open the "giphyintegrator.java" file then

```
giphyintegrator.java
8 import java.io.IOException;
12
13 public class giphyintegrator {
14     //authenticating and receiving images.
15 public String getimages(String keyword,int limit,int offset) throws IOException{
16
17     String err = "Error";
18
19     OkHttpClient client = new OkHttpClient();
20 //api key = dc6zaT0xFJmzC
21     Request request = new Request.Builder()
22         .url("http://api.giphy.com/v1/gifs/search?api_key=dc6zaT0xFJmzCq="+keyword+"&offset="+offset+"&")
23         .get()
24         .build();
25
26     Response response = client.newCall(request).execute();
27
28     if (!response.isSuccessful())
29     {
30         return err;
31     }
32
33     else
34     return response.body().string();
35 }
```

API KEY

replace the API key present inside the red box with your new API key.

Step-2: USING API-KEY TO CALL FUNCTIONS

Providing Parameters: GET requests should provide their parameters as a query string, part of the URL. API key should also be provided with other parameters as a part of the URL.

JSON Responses: All responses are sent JSON-encoded.

METHOD: GET SEARCH RESULT:

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.

URL: <http://api.giphy.com/v1/gifs/search>

Method : GET

Parameters : paramName="api_key", value="results offset, defaults to 0"

paramName="q", value="search query term or phrase"

paramName="limit", value="number of results to return"

paramName="offset", value="results offset, defaults to 0"

Output:

```
{
  "data": [
    {
      type: "gif",
      id: "FiGiRei2ICzzG",
      url: "http://giphy.com/gifs/funny-cat-FiGiRei2ICzzG",
      bitly_gif_url: "http://gph.is/1fIdL0l",
      bitly_url: "http://gph.is/1fIdL0l",
      embed_url: "http://giphy.com/embed/FiGiRei2ICzzG",
      username: "",
      source: "http://tumblr.com",
      rating: "g",
      caption: "",
      content_url: "",
      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"
    }
}

... 24 more items

],
"meta": {
    "status": 200,
    "msg": "OK"
},
"pagination": {
    "total_count": 1947,
    "count": 25,
    "offset": 0
}
}
```

Code:

The following code gives a JSON Output which contains information about Searched keyword.

```
package com.teamchat.integration.giphy.bot;

/* @author:Anuj Arora */

import java.io.IOException;

public class giphyintegrator {
    //authenticating and receiving images.
    public String getimages(String keyword,int limit,int offset) throws IOException{

        String err = "Error";

        OkHttpClient client = new OkHttpClient();
        //api key = dc6zaT0xFJmzC
        Request request = new Request.Builder()
            .url("http://api.giphy.com/v1/gifs/search?api_key=dc6zaT0xFJmzC&q="+keyword+"&offset="+offset+"&limit="+limit)
            .get()
            .build();

        Response response = client.newCall(request).execute();

        if (!response.isSuccessful())
        {
            return err;
        }

        else
        return response.body().string();
    }
}
```

Step-3: Exporting the WAR file

Step-4: Deploying the WAR file

Step-5: Ready to use

Step-6: Follow the commands given below to use Giphy:

Commands	Work
search	Type this command to search any gif image
next	Type this command to check the next result page for your searched keyword
help	It will tell user about the commands he/she should know before using

	giphy
rename	Type this command to change the name of the tags currently stored

Note:-

1. Getters and Setters classes are used for extracting the data from json object. These classes are in the package “com.teamchat.integration.giphy.classes”.

- **Datum**
- **Giphyd (main class for calling functions of other classes)**
- **Downsized**
- **DownsizedLarge**
- **DownsizedStill**
- **FixedHeight**
- **FixedHeightDownsampled**
- **FixedHeightSmall**
- **FixedHeightSmallStill**
- **FixedHeightStill**
- **FixedWidth**
- **FixedWidthDownsampled**
- **FixedWidthSmall**
- **FixedWidthSmallStill**
- **FixedWidthStill**

- **Images**
- **Meta**
- **Original**
- **OriginalStill**
- **Pagination**

2. Libraries used:-

- **gson 2.3.1 (for parsing json object)**
- **okhttp 2.4.0 (for authentication)**
- **okio 1.4.0 (prerequisite for okhttp)**
- **Teamchat Client SDK 1.5**

3. What changes should be done before running this project on any server?

- **Changes should be made in “giphy-config.properties” file , i.e your botemail, botpass, dbname, dbpass and tablename should be changed according to your requirement**
- **Changes should be made in “web.xml” file , i.e your bot-email, bot-password, db-host, db-name, db-user and db-password as per your requirement**
- **Table should be made in the same server in**

which you will deploy your project's WAR file

SQL query:-

```
use Bot;  
CREATE TABLE giphy_auth (emailid VARCHAR(150) NOT NULL,  
tag VARCHAR(32) NOT NULL,  
url VARCHAR(400) NOT NULL  
);
```