



Twitgoo Developer API Guide

April 15, 2009

Table of Contents

INTRODUCTION.....	1
USING THE DOCUMENTATION	1
GETTING STARTED.....	2
<i>Overview</i>	<i>2</i>
Requirements.....	2
Options.....	2
<i>Request and Response Formats.....</i>	<i>3</i>
REST Request Format	3
REST Response Format	3
<i>Parameters.....</i>	<i>4</i>
<i>Using Twitgoo Branding and Logos.....</i>	<i>4</i>
<i>Error Codes.....</i>	<i>4</i>
ACTIONS	5
<i>ping</i>	<i>5</i>
<i>uploadAndPost.....</i>	<i>6</i>
<i>upload</i>	<i>7</i>
<i>message info.....</i>	<i>8</i>
ADDITIONAL OPTIONS.....	9
<i>Image Redirects</i>	<i>9</i>
<i>Pre-populate Options</i>	<i>9</i>
TERMS OF USE	10

Introduction

The Twitgoo API allows you to create an interface to "talk" to the Twitgoo web site. This means that users do not have to leave your site to perform certain actions on the Twitgoo site, such as:

- Uploading media, and posting a tweet.
- Uploading media.

Using the Documentation

The documentation for the Twitgoo API is provided as a PDF document. It contains the following sections:

Topic/Book	Description
Introduction	An overview of the Twitgoo API.
Getting Started	An overview of requirements that must be met before using the Twitgoo API, including: <ul style="list-style-type: none">• An explanation of request and response formats.• Parameter definitions.• Error code definitions.• An overview of how to use Twitgoo branding on your site.
Actions	A list of the available actions in the API.
Terms of Use	The Terms of Use for using the API.

Getting Started

Overview

The Twitgoo API consists of a set of callable methods. All request formats take a list of named parameters. The parameters and example requests and responses are listed for each action. A list of actions can be found in the [Actions](#) topic.

Note: The Twitgoo API exposes identifiers for media and other uniquely identifiable objects. The format of the IDs can change over time, so relying on the current format may cause problems in the future.

Requirements

Before using the API, you must:

1. Understand the action parameters. See [Parameters](#).
2. Understand the REST request and response format. See [Request and Response Formats](#).
3. Determine which requests you want to send. Occasionally, a request requires data from a previous response.
4. Understand the error codes. See [Error Codes](#).

Important! For more information about the documentation, see [Using the Documentation](#).

Options

You can use Twitgoo branding and logos when you use the Twitgoo API, if you choose. See [Using Twitgoo Branding & Logos](#) for details.

Request and Response Formats

The Twitgoo API uses Representational State Transfer (REST) for requests and responses, so that domain-specific data can be transferred over HTTP without an additional messaging layer such as SOAP or session tracking via HTTP cookies. For an explanation of REST conventions, see <http://microformats.org/wiki/rest/urls>.

REST Request Format

API calls can be made via a POST or a GET, with the exception of handling uploads. Uploads must be sent via a multipart POST.

To request the Twitgoo service, invoke like this:

```
http://twitgoo.com/api/uploadAndPost
```

By default, REST requests send a REST response.

REST Response Format

The REST response format is a simple XML block. When using the REST request method, the response defaults to REST.

Successful Response

A successful response has a status "OK" and contains the return value for the call.

```
<?xml version="1.0" ?>
<rsp status="ok">
  <statusid>4444444</statusid>
  <userid>88888888888</userid>
  <mediaid>e583</mediaid>
  <mediaurl>http://twitgoo.com/e583</mediaurl>
  <imageurl>http://i38.tinypic.com/fc382.jpg</imageurl>
  <thumburl>http://i38.tinypic.com/fc382_th.jpg</thumburl>
</rsp>
```

Failure Response

A failure response has a status of "FAIL", and an error code is provided. See [Error Codes](#) for a list of error codes.

```
<?xml version="1.0" ?>
<rsp status="fail">
  <err code="1001" msg="Invalid twitter username or password" />
</rsp>
```

Parameters

The following table lists the most common parameters that are used in the actions. Other parameters may also be used and are discussed in each action.

Parameter	Description
username	Twitter username
password	Twitter password
media	Binary image data
message	Message post to Twitter. (URL will be added by Twitgoo)

Using Twitgoo Branding and Logos

When you use the Twitgoo logo, you must use it AS IS. The following requirements must be met:

- Nothing should ever be added to the logo, i.e., extra copy or images.
- Nothing should ever be removed from the logo.

The Twitgoo logo can be resized. Logos are provided upon request to support@twitgoo.com.

Error Codes

The following table lists the error codes that may be returned in a response.

Name	Error code	Reason
Bad Authorization	1001	Invalid twitter username or password
Bad Image	1002	Image not found
Bad Type	1003	Invalid image type
Bad File	1004	Image size is greater

Actions

All the actions and parameters use alpha characters.

ping

Ping Twitgoo to see if the system is up or is in maintenance mode.

REST Method

/api/ping

Request Parameters

None.

Response Parameters

Parameter	Description
status	OK or FAIL.

Example Request

The following request pings the Twitgoo server.

```
http://twitgoo.com/api/ping
```

Example Response

```
<?xml version="1.0" ?>  
<rsp status="ok">  
  <message>Pong</message>  
</rsp>
```

uploadAndPost

Upload media to the Twitgoo site, and post a status update to Twitter.

REST Path

/api /uploadAndPost

Request Parameters

Parameter	Optional	Description
username	N	Twitter username
password	N	Twitter password
media	Y	Binary image data
message	N	Message post to Twitter. (URL will be added by Twitgoo)
upload_url	Y	Source URL to upload

Response Parameters

Parameter	Description
status	OK or FAIL.
result	Contains the return value for the call or an error code. Expected results: <ul style="list-style-type: none">o statusid – Twitter status ido userid – Twitter user ido mediaid – Twitgoo media ido mediaurl – Twitgoo media URLo imageurl – Full-size image URLo thumburl – Thumbnail image URL

Example Request

The following request uploads media and posts a status update to Twitter.

```
POST http://twitgoo.com/api/uploadAndPost
```

Example Successful Response

A successful response looks similar to:

```
<?xml version="1.0" ?>
<rsp status="ok">
  <statusid>4444444</statusid>
  <userid>8888888888</userid>
  <mediaid>e583</mediaid>
  <mediaurl>http://twitgoo.com/e583</mediaurl>
  <imageurl>http://i38.tinypic.com/fc382.jpg</imageurl>
  <thumburl>http://i38.tinypic.com/fc382_th.jpg</thumburl>
</rsp>
```

Example Error Response

An unsuccessful response looks similar to:

```
<?xml version="1.0" ?>
<rsp status="fail">
  <err code="1001" msg="Invalid twitter username or password" />
</rsp>
```

upload

Upload media to the Twitgoo site

REST Path

/api /upload

Request Parameters

Parameter	Optional	Description
username	N	Twitter username
password	N	Twitter password
media	Y	Binary image data
upload_url	Y	Source URL to upload

Response Parameters

Parameter	Description
status	OK or FAIL.
result	Contains the return value for the call or an error code. Expected results: <ul style="list-style-type: none">o mediaid – Twitgoo media ido mediaurl – Twitgoo media URLo imageurl – Full-size image URLo thumburl – Thumbnail image URL

Example Request

The following request uploads media.

```
POST http://twitgoo.com/api/upload
```

Example Successful Response

A successful response looks similar to:

```
<?xml version="1.0" ?>
<rsp status="ok">
  <mediaid>e583</mediaid>
  <mediaurl>http://twitgoo.com/e583</mediaurl>
  <imageurl>http://i38.tinypic.com/fc382.jpg</imageurl>
  <thumburl>http://i38.tinypic.com/fc382_th.jpg</thumburl>
</rsp>
```

Example Error Response

An unsuccessful response looks similar to:

```
<?xml version="1.0" ?>
<rsp status="fail">
  <err code="1001" msg="Invalid twitter username or password" />
</rsp>
```

message info

Get the Twitter tweet ID, image URL, and thumb URL for a specific media ID.

REST Path

/api /message/info/[gooid]

Request Parameters

None.

Response Parameters

Parameter	Description
status	OK or FAIL.
result	Contains the return value for the call or an error code. Expected results: <ul style="list-style-type: none">o username – Twitter usernameo mediaid – Twitgoo media IDo imageurl – Full-size image URLo thumburl – Thumbnail image URLo statusid – Twitter tweet ID.

Example Request

The following request gets information about a specific Twitgoo ID.

```
GET http://twitgoo.com/api/message/info/e583
```

Example Successful Response

A successful response looks similar to:

```
<?xml version="1.0" encoding="UTF-8"?>
<rsp status="ok">
  <username>joenoia</username>
  <mediaid>3t0</mediaid>
  <mediaurl>http://twitgoo.com/</mediaurl>
  <imageurl>http://i39.tinypic.com/2rc2731.jpg</imageurl>
  <thumburl>http://i39.tinypic.com/2rc2731_th.jpg</thumburl>
  <statusid>1523967214</statusid>
</rsp>
```

Example Error Response

An unsuccessful response looks similar to:

```
<?xml version="1.0" ?>
<rsp status="fail">
  <err code="1001" msg="Invalid twitter username or password" />
</rsp>
```

Additional Options

Other options are available for getting data.

Image Redirects

The following options are available for redirecting to a specific Twitgoo image:

`/show/(thumb|mini|img)/[goid]`

`/[goid]/(thumb|mini|img)` – Redirects to full URL of given size (thumb and mini are identical; img is the full size)

The image redirect strings on the homepage looks similar to:

`http://twitgoo.com/show/thumb/49d`

`http://twitgoo.com/49d/thumb`

Pre-populate Options

On the homepage (/) you can GET the page with the following pre-population parameters:

`"tweet-text"` - Text to tweet

`"twitter_in_reply_to_status_id"` - In reply to a twitter status

The pre-populate string on the homepage looks similar to:

`http://twitgoo.com/?tweet-text=OMG`

Terms of Use

Twitgoo provides the materials in this help system on an "as is" and "as available" basis, and makes no representations or warranties as to the accuracy, completeness, or quality of these materials, nor that they will meet any of your particular needs or objectives. Furthermore, code samples in this help system are provided as a development resource and may or may not work in your particular situation. Twitgoo disclaims any warranties implied by law, and reserves the right to change these materials without notice. Your use of these materials is at your own risk and implies that you accept the Twitgoo *Legal & Privacy* at <http://Twitgoo.com/privacy.php> and the Twitgoo *Terms of Service* at <http://Twitgoo.com/terms.php>.

Access is at the discretion of Twitgoo, and there is no guarantee of service. If the terms are violated, your access may be removed without warning.