WhatsApp Messaging API Specification V1.01

2nd July, 2024

Contents

Introduction

This document aims at providing all the information you need to work with our WhatsApp Messaging API.

Scope

This document covers the various methods where each has endpoints, request parameters, API response and sample code (requests and responses). This document may be used to guide the development of custom-built applications to access the API.

Audience

This document is intended for Institutions interested in sending WhatsApp messages via our API. Such messages may include but not limited to: Text (also with preview URL), Images, Audio, Documents, Stickers, Videos, Contacts, and Location. In all of these scenarios, the message can also be sent by URL.

Notice/Disclaimer

Changes may be made periodically to the information in this document. Such changes will be incorporated in new editions of the specification.

Operations

The API provides various operations or methods to be used to submit or fetch information.

Using the API

Request Submission URL

All requests to the API system must be submitted to the following URL: https://www.socnetsolutions.com/projects/bulk/whatsapp/api.php

The webservice is a JSON based web service with a couple of webservice methods that will ONLY be accessed via the URL above. The webservice methods include the Authentication methods and Message sending methods.

Prerequisites

To use the API, you must, first of all, have a **Socnet API User Account**. The API is currently ONLY available for Institution Accounts and NOT individuals. Once you have an Institution Account, you must obtain the **API credentials**. You will be able to login into the interface at:

https://www.socnetsolutions.com/projects/api/

From this interface, you can already test Whatsapp Messaging service by going to View->WhatsAPp on the menu provided as shown in Figure 1. You may also obtain the API credentials from this web interface of your

User Account. The API credentials are used to map all your API requests to your User Account, and for other security purposes.

Please note that during the test phase, test credits may be obtained from a Socnet user with administrative rights.

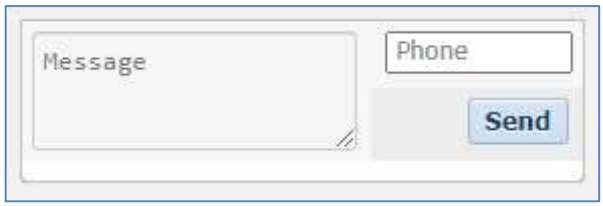

Figure 1

General Request Format

```
{
        "api_key": "32-byte API key here",
        "msisdn": ""
}
```

General Response Format

```
{
        "message": "response_message",
        "status": response_status_code
}
```

Description of the Response

Response Parameter	Type	Presence	Description
message	Str	O	Description of the response. Often empty for successful requests and a description of the error if request has failed
Status	Int/ Str	O	Status of the request

Status Codes

For every response, an HTTP Status Code will be added. The most common codes to be expected include the following:

Status Code	Description
0	OK – The operation was successful. This is the standard success code and default option.

1	FAIL – The operation was not successful. Something wrong with the request.
401	Invalid user credentials
403	Insufficient credit on account
404	There is no text in the image
500	Internal server error – If something unexpected breaks.
503	Service unavailable – If the endpoints cannot be reached.

Request Parameter status

ABBRV	Status	Definition
M	Mandatory	Mandatory parameters must be supplied
C	Conditional	Conditional parameters must be supplied when applicable rule is met if not supplied, request will be not be processed.
O	Optional	This can be supplied when data is available no rejections here
Str	String	Must be submitted with double quotes
Int	Integer	The fields accept only numeric values
Dec	Decimal	Numeric data with decimal point allowed after the character
AN	Alphanumeric	Letters and numbers
Bin	Binary	Binary data for images and/or documents

API Operations

Authentication

Requests made to our APIs must be authenticated by supplying the provided API Key:

Authenticating using the API Key

```
{
        'api_key': '32-byte API key here'
}
```

Successful Response using API username and password

```
{ “status”: “valid” }
```

Failed Response – Wrong credentials

```
{
        "message": "Invalid user credentials",
        "status": 401,
}
```

Failed Response – Insufficient credits

```
{
        "message": "Insufficient credit (0) for transaction charge of xxx",
        "status": 403,
}
```

Failed Response – Missing parameters

```
{
        "data": null,
        "message": "The [parameter] field is required."
        "status": 400
}
```

Messaging Operations

General Success Response

```
{
        "message": "[Resource name] submitted successfully.",
        "status": 0
}
```

General Failure Response

Missing parameter

```
{
        "message": "The [parameter] field is required."
        "status": 422
}
```

Wrong format

```
{
        "message": "The [parameter] field must be [format]."
        "status": 422
}
```

Server error

```
{
        "message": "[Failure reason]",
        "status": 500
}
```

Method not allowed

```
{
        "message": "Method is not allowed for the requested route",
        "status": 405
}
```

Resource not found

```
{
        "data": null,
        "message": "[Resource] not found",
        "status": 404
}
```

Message Sending

This method sends a WhatsApp message to the provided phone number. Before which, the method checks whether the provided phone number is registered on WhatsApp or not.

Request Format

```
{
        'api_key': '32-byte API key here',
        "msisdn": "",
        "msg": ""
}
```

Request parameters

Parameter	Type	Mandatory/Optional	Definition
msisdn	N12	M	Start with 256
msg	Str	M	Text message to send
template	Str	O	Name of approved template (if any)

*Note that in case a One Time Password (OTP) template is used, then the msg should contain only the OTP and no other text. The rest of the text will be generated automatically by the API, say, if msg contains "1234" then the user receives: "**1234** is your verification code. For your security, do not share this code." Perhaps together with a "Copy code" button for the user to copy and paste the code into their parent application.*

Response Data parameters

Response Parameter	Status	Definition
status/sent	Str	Status of the request
messaging_product	Str	Messaging gateway
contacts	N12	MSISDN sent to
messages	Str	Message ID and status
cost	Dec	Charge applied to SUCCESSFUL request/operation

Success Response – channel 1


```
{
        "sent": "true",
        "from": "256701234567@c.us",
        "message_type": "chat",
        "messageId": "3EB002BAE9545124A5B7B2",
        "cost": "90"
}
```


Success Response – channel 2


```
{
        "messaging_product": "whatsapp",
        "contacts": [{"input": "256701234567","wa_id": "256701234567"}],
        "messages": [{"id": "message_id", "message_status": "accepted"}],
        "cost": 0
}
```


The API will automatically choose one of the two channels to use depending on the traffic and throughput of the channel. So, expect any of the two responses above.

Unregistered Phone Number Response


```
{
        "status": "invalid"
}
```