**User Guide for WhatsApp Demo**

Table of Contents

[A. WhatsApp Demo API 3](#_Toc178585353)

[1. Introduction 3](#_Toc178585354)

[2. API 3](#_Toc178585355)

[3. Prerequisites 3](#_Toc178585356)

[4. Authentication Setup 3](#_Toc178585357)

[5. API Endpoints 4](#_Toc178585358)

[5.1 Send Text Message API 4](#_Toc178585359)

[5.2 Send Image API 5](#_Toc178585360)

[5.3 Send Video API 6](#_Toc178585361)

[5.4 Send Audio API 7](#_Toc178585362)

[5.5 Send Document API 9](#_Toc178585363)

[5.6 Send Location API 10](#_Toc178585364)

[6. Error handling 11](#_Toc178585365)

[7. Rate Limiting 11](#_Toc178585366)

[8. Conclusion 12](#_Toc178585367)

[B. WhatsApp Installation 13](#_Toc178585368)

[1) Flowchart 13](#_Toc178585369)

[2) Draw.io 14](#_Toc178585370)

[1. Introduction 14](#_Toc178585371)

[2. Features 14](#_Toc178585372)

[3. WhatsApp Installation Flowchart using Draw.io 14](#_Toc178585373)

[3.1 Draw.io 14](#_Toc178585374)

[3.2 Create New Diagram 14](#_Toc178585375)

[3.3 Navigation 14](#_Toc178585376)

[3.4 Start 14](#_Toc178585377)

[3.5 Android/iOS 14](#_Toc178585378)

[3.6 Installation 15](#_Toc178585379)

[3.7 End 15](#_Toc178585380)

# WhatsApp Demo API

## Introduction

This guide will assist the user in understanding the Dummy API, which functions similarly to WhatsApp and can be used for sending and receiving messages, managing multimedia files, and monitoring the status of messages, and much more.

## API

API stands for Application Programming Interface. It acts like a messenger that receives your requests, relays them to another system and then returns the response.

**Why to use the WhatsApp Dummy API?**

There are some specific reasons to use the dummy API which are as follows:

* **Messaging**: You can send text messages to anyone.
* **Scalable:** It can handle large volumes of messages whether in a single chat or in a group.
* **Security:** All theAPI requests are secured using token-based authentication, ensuring that your communication is safe and private. They use 2-way authentication system.
* **Rich Multimedia Support:** It enhances the user experience by allowing to API to send and receive high-quality multimedia content, such as images, videos, audio files and documents, etc.

## Prerequisites

Some of the prerequisites for the Demo API to work are:

* **API key**: The user needs an API key to authenticate the requests. It should not be shared with anyone.
* **Internet Access**: The system should have proper connectivity.
* **Postman**: This tool helps the user to test the APIs.
* **Text Editor**: The user can use any text editor tools for writing codes.

## Authentication Setup

1. **Obtain the API Key**: When the user needs to setup the dummy API, they will receive one API key. It is a long string having some numbers and letters combined, which can not be shared with anyone.

|  |
| --- |
| API\_KEY = "abcdef1234567890" |

1. **Postman Tool**: You will use the Postman to test the APIs provided.

## API Endpoints

### Send Text Message API

This API sends a text message to a specified phone number.

**API Details**

|  |  |
| --- | --- |
| **Type** | **Value** |
| API Signature | POST /messages/send |
| Method | POST |

**Header details**

|  |  |
| --- | --- |
| **Header Name** | **Header Value** |
| Accepts | Application/json |
| Content Type | Application/json |

**Request Details:**

|  |
| --- |
| {  "to": "+1234567890",  "message": "Hello, this is a test message.",  "message\_template\_id": "temp\_12345" // optional  } |

**Response Details:**

|  |
| --- |
| {  "status": "success",  "message\_id": "msg\_12345",  "timestamp": "2024-09-28T10:23:00Z"  } |

**Parameter Details:**

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Value** | **Description** |
| to | +1234567890 | |  | | --- | |  |  |  | | --- | | The recipient's phone number. | |
| message | "Hello, this is a test message." | The text message content. |
| message\_template\_id | temp\_12345 | Template ID for preformatted messages. |

### Send Image API

This API sends an image message to a specified phone number with an optional caption.

**API Details**

|  |  |
| --- | --- |
| **Type** | **Value** |
| API Signature | POST /media/sendImage |
| Method | POST |

**Header details**

|  |  |
| --- | --- |
| **Header Name** | **Header Value** |
| Accepts | Application/json |
| Content Type | Application/json |

**Request Details:**

|  |
| --- |
| {  "to": "+1234567890",  "image\_url": "https://example.com/sample.jpg",  "caption": "Check this out!"  } |

**Response Details:**

|  |
| --- |
| {  "status": "success",  "media\_id": "media\_67890",  "message\_id": "msg\_67890",  "timestamp": "2024-09-28T10:25:00Z"  } |

**Parameter Details:**

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Value** | **Description** |
| to | +1234567890 | |  | | --- | |  |  |  | | --- | | The recipient's phone number. | |
| Image\_url | https://example.com/sample.jpg | The URL of the image to be sent. |
| caption | Check this out! | Optional caption for the image. |
| Message\_id | msg\_67890 | The unique identifier of the message sent. |

### Send Video API

This API sends a video message to a specified phone number.

**API Details**

|  |  |
| --- | --- |
| **Type** | **Value** |
| API Signature | POST /media/sendVideo |
| Method | POST |

**Header details**

|  |  |
| --- | --- |
| **Header Name** | **Header Value** |
| Accepts | Application/json |
| Content Type | Application/json |

**Request Details:**

|  |
| --- |
| {  "to": "+1234567890",  "video\_url": "https://example.com/sample.mp4",  "caption": "Here's a cool video!"  } |

**Response Details:**

|  |
| --- |
| {  "status": "success",  "media\_id": "media\_34567",  "message\_id": "msg\_34567",  "timestamp": "2024-09-28T10:26:00Z"  } |

**Parameter Details:**

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Value** | **Description** |
| to | +1234567890 | |  | | --- | |  |  |  | | --- | | The recipient's phone number in international format. | |
| video\_url | https://example.com/sample.mp4 | The url of the video to be sent. |
| caption | Here's a cool video! | Optional caption for the video. |

### Send Audio API

This API sends an audio message to another user.

**API Details**

|  |  |
| --- | --- |
| **Type** | **Value** |
| API Signature | POST /media/sendAudio |
| Method | POST |

**Header details**

|  |  |
| --- | --- |
| **Header Name** | **Header Value** |
| Accepts | Application/json |
| Content Type | Application/json |

**Request Details:**

|  |
| --- |
| {  "to": "+1234567890",  "audio\_url": "https://example.com/sample.mp3"  } |

**Response Details:**

|  |
| --- |
| {  "status": "success",  "media\_id": "media\_45678",  "message\_id": "msg\_45678",  "timestamp": "2024-09-28T10:27:00Z"  } |

**Parameter Details:**

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Value** | **Description** |
| to | +1234567890 | |  | | --- | |  |  |  | | --- | | The recipient's phone number. | |
| Audio\_url | https://example.com/sample.mp3 | The url of the file audio to be sent. |

### Send Document API

This API sends any document(pdf or word) to a specified user.

**API Details**

|  |  |
| --- | --- |
| **Type** | **Value** |
| API Signature | POST /media/sendDocument |
| Method | POST |

**Header details**

|  |  |
| --- | --- |
| **Header Name** | **Header Value** |
| Accepts | Application/json |
| Content Type | Application/json |

**Request Details:**

|  |
| --- |
| {  "to": "+1234567890",  "document\_url": "https://example.com/file.pdf",  "filename": "file.pdf"  } |

**Response Details:**

|  |
| --- |
| {  "status": "success",  "media\_id": "media\_98765",  "message\_id": "msg\_98765",  "timestamp": "2024-09-28T10:28:00Z"  } |

**Parameter Details:**

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Value** | **Description** |
| to | +1234567890 | |  | | --- | |  |  |  | | --- | | The recipient's phone number. | |
| Document\_url | https://example.com/file.pdf | The URL of the document to be sent. |
| filename | File.pdf | The name of the file. |

### Send Location API

This API sends current or live location to another specified user.

**API Details**

|  |  |
| --- | --- |
| **Type** | **Value** |
| API Signature | POST /messages/sendLocation |
| Method | POST |

**Header details**

|  |  |
| --- | --- |
| **Header Name** | **Header Value** |
| Accepts | Application/json |
| Content Type | Application/json |

**Request Details:**

|  |
| --- |
| {  "to": "+1234567890",  "latitude": "40.712776",  "longitude": "-74.005974",  "name": "New York City",  "address": "New York, USA"  } |

**Response Details:**

|  |
| --- |
| {  "status": "success",  "message\_id": "msg\_54321",  "timestamp": "2024-09-28T10:35:00Z"  } |

**Parameter Details:**

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Value** | **Description** |
| to | +1234567890 | |  | | --- | |  |  |  | | --- | | The recipient's phone number. | |
| latitude | 40.712776 | Latitude of the location. |
| longitude | 74.005974 | Longitude of the location. |
| name | New York City | Name of the location. |
| address | New York, USA | Address details. |

## Error handling

These API provides the standardized error responses for client and server-side errors.

|  |  |  |
| --- | --- | --- |
| **Error Code** | **Message** | **Description** |
| 400 | Bad request | The request is missing required parameters. |
| 401 | Unauthorized | Invalid or missing API key. |
| 403 | Forbidden | Access to the requested resource is forbidden. |
| 404 | Not found | The requested resource was not found. |
| 429 | Too many Requests | The rate limit has been exceeded. |
| 500 | Internet Server Error | An unexpected error occurred on the server. |

## Rate Limiting

The API enforces the rate limits. If the limit is exceeded, the API will return a 429 Too Many Requests Error.

**Rate Limit**: 100 requests per minute.

**Error Response (429)**:

|  |
| --- |
| {  "status": "error",  "error\_code": 429,  "error\_message": "Too many requests, try again later."  } |

## Conclusion

This comprehensive API allows for robust messaging functionality with support for media like image, audio, video, etc. This guide helps to integrate messaging services into any application seamlessly.

# WhatsApp Installation

## Flowchart

A screenshot of a phone

Description automatically generated

## Draw.io

### Introduction

Draw.io is a free online tool where users can build a wide range of diagrams like flowcharts, charts, network diagrams and more. It can be used by any person since there is no use of technicality in designing any diagrams with the help of a huge number of shapes and templates.

### Features

* Web based Platform – The application is web based where the installation is not required. The user can easily start designing after opening the draw.io browser.
* Interface – The interface is simple where the user can use the drag and drop option to create their diagrams.

### WhatsApp Installation Flowchart using Draw.io

#### Draw.io

1. Open any browser and search for [Draw.io](https://app.diagrams.net/).
2. You can create a new diagram from scratch or else you can open an existing diagram.

#### Create New Diagram

1. Select ‘Create New Diagram’ .
2. Once you choose, a new blank template will open in front of you. And you will have the option to select from the different templates. You can rename the file to WhatsApp Installation.

#### Navigation

1. The different toolbars with various shapes are present on the left side of the template. And the template view option is present on the right side.
2. In this flowchart, you can find out that the circle shape is used for Start and End position, then rectangle shape is used for Processes and lastly the diamonds are used for decisions.

#### Start

1. You can drag and drop a circle from the toolbox.
2. You need to double click the circle so that you can produce text.
3. Label it as Start.

#### Android/iOS

1. Draw a diamond and label it as Android/iOS to check whether the phone is Android or iOS.
2. Creating branches for Android and iOS users.
3. If it is an Android phone, you should go in the left side to open the Google play store application, which is in the rectangle box and labeled as ‘Google Play Store’.
4. If it is an iOS phone, you should go to the right side to open the Apple App Store application, which relates to the rectangle box by the arrow and labelled as ‘Apple App Store’.
5. Search for the WhatsApp Messenger from both path of Android and iOS.
6. Draw a diamond and label it as ‘Is WhatsApp found?’. It will check whether the application is there in the application Store.
7. If the answer is no, then you need to check the internet connection.

#### Installation

1. If it is Yes, then select the WhatsApp Messenger, which is labelled on the rectangle box.
2. You can click on the Install or Get button.
3. You can create a diamond where the label should be ‘Is the App downloading’. So, it will check whether the application is downloading on the phone or not. If not, then again you need to restart your process because your internet connection can be low. If yes, then wait for the installation to happen. Once it is done, you will receive a notification that the Application is Installed.
4. In the rectangle box, the label mentioned ‘Open WhatsApp’. So , you can open the application and agree to the terms and conditions which will pop on the screen.
5. Once the T&C is done, it will ask for the verification of the phone number by sending you a code. A diamond has been created to check whether the verification code is received or not.
6. If the code is not received, then again, the process will repeat from the verification phase. If you receive the code, type in the blanks or else it will automatically capture the code.
7. After the verification is over, you can set up the profile and the installation are complete.

#### End

1. You can drag and drop a circle from the toolbox.
2. You need to double click the circle so that you can produce text.
3. Label it as End.