

# Karix RCM API for WhatsApp

Date: 28/05/2020

## Table of Contents

Introduction .....	4
Terms & Abbreviations .....	4
About WhatsApp Business Messaging.....	4
How Your Business Appears to Users .....	4
Getting Verified with WhatsApp.....	5
Sending notifications .....	5
Conversational Messaging on WhatsApp .....	5
Customer Care Window .....	5
Discovery.....	6
Deep link format .....	6
Click to Chat Web Link .....	6
Formatting in WhatsApp messages .....	6
Karix RCM API Specification .....	6
Prerequisites .....	6
Account with Karix .....	6
Opt-In Requirement.....	7
Using WhatsApp numbers .....	7
Features supported by this API.....	7
Sample API Formats .....	7
API endpoint .....	7
Request headers .....	7
JSON Payload Parameters.....	8
Generic Payload .....	12
Sending a Messages to WhatsApp (One to One).....	13
Sample request for Text message.....	13
Sample request for Auto-Template message .....	14
Sample request for Template message .....	15
Sample request for Media template.....	16
Sample request for quick reply button .....	19
Sample request for call to action .....	21
Sample request for video.....	23
Sample request for Media message .....	24
Sample request for Media message (stickers).....	25
Supported Mime Types.....	26

Sample request for sending Location .....	27
Sample response format .....	27
Status Code & Descriptions.....	28
Fallback SMS .....	29
Fallback conditions:.....	29
Sample Payload .....	29
Webhooks .....	30
Types of WhatsApp webhook events Supported by RCM API.....	30
Delivery events.....	30
Error code & descriptions .....	33
User initiated events .....	33
Guidelines .....	39

## Introduction

This guide provides specifications of Karix RCM (Rich Communication messaging) API for WhatsApp, which enables brands to reach over 1.5billion users on WhatsApp using a Simple REST API. This guide is intended for developers who want to write application that can interact with Karix Rich content messaging services.

This guide assumes that the user is familiar with and have experience in applying the following technologies and concepts:

- HTTP/HTTPS communications using the GET and POST methods for parameter passing
- Any of the programming language such as Java, C/C++, Visual Basic or PHP

## Terms & Abbreviations

Terms	Descriptions
<b>RCM</b>	Rich Communication messaging is a next-generation messaging protocol that aims to unifying communication across all popular messaging apps. With RCM, businesses can engage users with branded messages, share rich content including cards, carousels for scrolling through products, suggest replies and actions such as adding appointments to a calendar or opening maps for directions to that appointment.
<b>Verified Business</b>	WhatsApp has verified that an authentic brand owns this account. A Verified account has a green checkmark badge in its profile. Very few businesses will be WhatsApp verified. Being verified on Facebook or Instagram will not help your business be WhatsApp verified.
<b>Regular</b>	An account that is using one of WhatsApp's business products but has not been confirmed nor verified by WhatsApp. A business account has a grey question mark badge in its profile.

## About WhatsApp Business Messaging

WhatsApp helps more than 1 billion people connect and share with the people they care about. WhatsApp is a simple, secure, and reliable way for businesses to reach their customers all over the world. This guide will help businesses on board and build their first official WhatsApp messaging experience using Karix RCM API.

### How Your Business Appears to Users

Depending on your number type, users will see different things. However, if a user has already saved the business number in their address book, the Verified Name will not take precedence; the name from the address book will always be displayed. The phone number will still be visible in the contacts view. Please be prepared for users who may try to call this number and have a plan for how to handle these incoming calls.

If the number is **verified**, the Verified Name will be visible in the chat list, chat screens, chat groups, and contact view instead of the phone number. There will be a green "**Verified**" checkmark in the contact view.

If the number is **confirmed**, the Verified Name will only be visible in the contact view. All other views will show the phone number. The contact view will show a grey "**Confirmed**" checkmark with the verified name in smaller text.

If it is a **Regular** number, the Verified Name will only be visible in the contact view, all other views will show the phone number. The contact view will show the account is a business account with the verified name in smaller text.

### Getting Verified with WhatsApp

By default, all numbers are in the **Regular** state. Once Karix has successfully enabled your number for WhatsApp, Karix will submit it to WhatsApp to be **verified**. WhatsApp will review the request and your business and determine which level to assign it. The Verified business account status is given after at least one month of go live release with an average traffic of at least 1000 messages per day for a 7-day aggregate.

### Sending notifications

WhatsApp requires that business-initiated Notifications sent by your application be templated and pre-registered, with exception of messages [] sent as a reply to a user-initiated message.

Both Text & Media content can be sent as a Notification message. Please contact our [provisioning team](#) / your account manager to register/configure/whitelist your message template.

**Message types:** TEMPLATE | AUTO\_TEMPLATE | MEDIA\_TEMPLATE

Karix RCM Api allows businesses to send interactive buttons in notification messages.

There are two types of predefined buttons offered:

- **Call-to-Action** - Allows your customer to call a phone number and visit a website
- **Quick Reply** - Allows your customer to return a simple text message

Buttons are subject to approval. Hence Media templates or Templates with buttons need to be shared with Karix prior, for registering and approval.

### Conversational Messaging on WhatsApp

To have a 2-way conversation with a user, you need to receive messages from them. Users can send you a message in response to a templated notification or directly

### Customer Care Window

A WhatsApp Customer Care Window begins with a user-initiated message to your WhatsApp number. Sessions are valid for 24 hours after the most recently received message, during which time you can communicate with them using free form messages. In order to send a message outside the 24-hour Session window, you must use a pre-approved template (see Sending Notifications section above).

**Message types:** TEXT | ATTACHMENT (Image/Audio/Video/stickers/document) | LOCATION | CONTACTS

## Discovery

You can have customers initiate a conversation with you on WhatsApp via URL schemes, embedded in web / mobile apps. If the user has WhatsApp iOS, Android or desktop installed, clicking the deep link will open a conversation with you inside the app.

## Deep link format

The following is the deep link format

**whatsapp://send?phone=<e164 number>&text=Hello!**

## Click to Chat Web Link

You can create your own link, use **https://wa.me/<number>** where the <number> is a full phone number in international format. Omit any zeroes, brackets or dashes when adding the phone number in international format. For a detailed explanation on international numbers, read this [article](#). Please keep in mind that this phone number must have an active account on WhatsApp.

**<https://wa.me/whatsappphonenumber?text=urlencodedtext>**

## Example:

**<https://wa.me/91XXXXXXXXXX?text=I'm%20interested%20in%20your%20car%20for%20sale>**

## Formatting in WhatsApp messages

WhatsApp allows text, emoji's and some formatting in messages. To format all or part of a message, use these formatting symbols:

Your messages can be **bold**, *italicized*, struck-through, or preformatted.

Formatting	Symbol	Example
Bold	Asterisk (**)	Your total is **₹1250/-**.
Italic	Underscore (_)	Welcome to _WhatsApp_!
Strike-through	Tilde (~)	This is ~better~ best!
Code / Pre-formatted	Three backticks (`)`)	```print 'Hello World';```

The text inside the formatting symbols will take on the format you choose.

## Karix RCM API Specification

### Prerequisites

To start using Karix RCM API

### Account with Karix

- Get an post-paid account with RCM media enabled created at Karix end
- Log in to ([www.karix.solutions/lounge/sign\\_in.php](http://www.karix.solutions/lounge/sign_in.php)) to generate Keys which is to be used during authentication

### Opt-In Requirement

- Brands to ensure user explicit opt-ins for the service/product notifications to deliver over WhatsApp
- Brands can gather user opt-ins via a Webpage or sign-up flow, SMS etc.
- Sending users message without an Opt –in May result in user blocking your business and suspension of your WhatsApp business account

### Using WhatsApp numbers

- To send Message on WhatsApp using Karix RCM API you will need a phone number as well
- Karix provisions WhatsApp numbers to brands (Number to be approved/register with WhatsApp)
- Users can avail pre-provisioned Karix WhatsApp numbers for any quick prototype.

### Features supported by this API

Karix RCM API is designed to be simple with features such as:

- Send One to One message to WhatsApp
  - Content Supported
    - Text
    - Image
    - Audio
    - Video
    - Sticker
    - Document
  - Buttons (Quick reply & calls to action)
  - Location
  - Contacts

### Sample API Formats

#### API endpoint

RCM API requests are made to the following server:

**`https://rcmapi.instaalerts.zone/services/rcm/sendMessage`**

### Request headers

The following is the description of request headers:

HTTP header	Description
<b>Authentication</b> required	<i>Bearer &lt;account key&gt;</i>
<b>Content-Type</b> required	application/json

## JSON Payload Parameters

The following is the parameters supported by the Karix RCM API:

	PARAMETERS	IS REQUIRED	TYPE	DEFAULT VALUE	DESCRIPTION
message			tag		
	channel	YES	string		<i>Channel ID alphanumeric This will be identifier to the exact endpoint (gateway) where enterprise client wishes to send message. For WhatsApp please use WABA</i>
	content	YES	tag		
	sender	YES	tag		
content					
	preview_url	NO	boolean		<i>By default, the mobile WhatsApp application recognizes URLs and makes them clickable. To include a URL preview, include "preview_url": true and make sure the URL begins with http:// or https://. A hostname is also required</i>
	type	YES	string		<i>one of the below : <b>TEXT</b> or <b>ATTACHMENT</b> or <b>TEMPLATE</b> or <b>LOCATION</b></i>
Inside "content" tag For type: TEXT/ AUTO-TEMPLATE					
	text	YES	string		<i>Some text content user want to send free form text content including WhatsApp formatting</i>
Inside "content" tag For type: ATTACHMENT					
	type	YES	string		<i>document/image/audio</i>
	mimeType	YES	string		<i><b>For Image</b> : image/png, image/jpeg <b>For document</b>: application/pdf <b>For audio</b>: audio/mpeg, audio/wav</i>
	caption	NO	string		<i>Some tile as a caption</i>
	attachmentData	YES	string		<i>string contains attachment binary data of Base64 encoded</i>



<b>Inside “content” tag For type: TEMPLATE</b>					
template					
	templateId	YES			Template ID configured for Whatsapp template(HSM) , this template ID to be configured in Karix
	parameterValues	YES	“key” : “value” Entries		Values should be given in the order like “0” : “value1”, “” : “value2” ....  In case no place holders in the template content, then pass the entries leaving blank parameterValues
<b>Inside “content” tag For type: MEDIA TEMPLATE</b>					
Media template					
	templateId	YES	string		This templateID denotes the media template ID
	bodyparameterValues	YES			Denotes media template message body. Values should be given in the order like “0” : “value1”, “” : “value2” ....
Media					
	Type	YES	string		Denotes the type of media template (Image/Document/Video/Buttons/ Actions)
	URL	YES	string		Denotes publicly accessible reference address
Buttons					
Quick replies					
	Index	YES	string		Values should be given in the order like “0”, or s“1”. It denotes the order of the button in the message.
	Payload	YES	object		Payload is which you will receive back in the callback when the user clicks on the quick reply button.
Actions					

	Type	YES	string		<i>Denotes the type of actions (Dial &amp; URL) to display</i>
	Index	YES	string		<i>Values should be given in the order like "0", or "1". It denotes the order of the actions in the message.</i>
	Payload	YES	object		<i>Payload is which you will receive back in the callback when the user clicks on the quick reply button.</i>
location					
	longitude	YES	String		<i>Longitude of the Location</i>
	latitude	YES	String		<i>Latitude of the Location</i>
	name	NO	String		<i>Name of the Location</i>
	address	NO	String		<i>Address of the Location. Only displayed if "name" is present</i>
recipient					
	to	YES	string		<i>Recipient whatsapp number like 919999999999 etc.</i>
	recipient_type	NO	string	individual	<i>Either individual or group</i>
reference					
	cust_ref	NO	string		<i>A unique id send by the customer for every message</i>
	messageTag1-messageTag5	NO	string		<i>To Specify the message tag</i>
sender					
	name	NO	string		<i>Descriptive name of Sender</i>
	from	YES	string		<i>Brands WhatsApp number</i>
smsFallback					
	sender	YES	string		<i>Sender ID of source (Alpha Sender IDs up to 6 characters).</i>

	destination	YES	string		<i>Destination no. to the fallback to be done</i>
	message	YES	string		<i>Message to be sent</i>
preferences					
	webHookDNId	NO	string		<i>This refers to the web hook rule ID configured at Karix, DN events will be postback to the specified rule id. (Please reach your account manager at Karix to register the webhook rule)</i>
metadata			tag		
	version	YES		v1.0.9	

## Generic Payload

Below is the Sample Generic Payload

JSON API Payload Format	<pre>{   "message":{     "channel":"Some Channel Identifier",     "content":{       "preview_url":false,       "type":"TEXT_or_ATTACHMENT_or_TEMPLATE",       "text":"Some Text in case content type is Text ",       "attachment":{         "type":"image/document",         "mimeType":"standard mime type as per the content type e.g.application / pdf",         "attachment_id":"attachment123",         "caption":"Some Attachment Text (Caption)",         "attachmentData":"&lt;&lt;&lt;&lt;SOME BASE 64 DATA&gt;&gt;&gt;&gt;"       },       "template":{         "templateId":"Some Template ID tid-1",         "parameterValues":{           "key1":"ValueA",           "key2":"ValueB"         }       }     },     "recipient":{       "to":"Some Phone Number/To",       "recipient_type":"individual",       "reference":{         "cust_ref":"Some Customer Ref",         "messageTag1":"Message Tag Val1..5",         "conversationId":"Some Optional Conversation ID"       }     },     "sender":{       "name":"Optional Name",       "from":"Sender WhatsApp Number "     },     "preferences":{       "webHookDNId":"WEB HOOK Configured ID to recieve DN "     },     "metaData":{       "version":" v1.0.9"     }   } }</pre>
-------------------------	---

## Sending a Messages to WhatsApp (One to One)

It refers to sending of single message to a single recipient

### Sample request for Text message

The Following sample code is an example of sending a text message to WhatsApp

API URL	<code>https://rcmapi.instaalerts.zone/services/rcm/sendMessage</code>
Request Headers	<b>Content-Type:</b> <code>application/json</code> <b>Authentication:</b> <code>Bearer &lt;account key&gt;</code>
Request Payload	<pre>{   "message":{     "channel":"WABA",     "content":{       "preview_url":false,       "text":"Some type of Text Message ",       "type":"TEXT"     },     "recipient":{       "to":"919999999999",       "recipient_type":"individual",       "reference":{         "cust_ref":"Some Customer Ref",         "messageTag1":"Message Tag Val1",         "conversationId":"Some Optional Conversation ID"       }     },     "sender":{       "from":"919900000000"     },     "preferences":{       "webHookDNId":"1001"     }   },   "metaData":{     "version":"v1.0.9"   } }</pre>

### Sample request for Auto-Template message

The following code provide an example of sending an Auto-Template Message.

API URL	<code>https://rcmapi.instaalerts.zone/services/rcm/sendMessage</code>
Request Headers	<b>Content-Type:</b> <code>application/json</code> <b>Authentication:</b> <code>Bearer &lt;Account key&gt;</code>
Request Payload	<pre>{   "message": {     "channel": "WABA",     "content": {       "preview_url": false,       "text": "Hi Rajasekar! Thanks for starting your personal training plan. We'll send you a weekly update with your new schedule. You can log-in online using your training ID 123456. Stay fit !",       "type": "AUTO_TEMPLATE"     },     "recipient": {       "to": "919999999999",       "recipient_type": "individual",       "reference": {         "cust_ref": "cust_ref123",         "messageTag1": "Message Tag 001",         "conversationId": "Conv_123"       }     },     "sender": {       "name": "whatsappdemo",       "from": "919900000000"     },     "preferences": {       "webHookDNid": "1001"     },     "metaData": {       "version": "v1.0.9"     }   } }</pre>

### Sample request for Template message

The following code provide an example of sending a Template Message, for creating templates please reach your account manager at Karix with complete Message content.

API URL	<code>https://rcmapi.instaalerts.zone/services/rcm/sendMessage</code>
Request Headers	<b>Content-Type:</b> <code>application/json</code> <b>Authentication:</b> <code>Bearer &lt;Account key&gt;</code>
Request Payload	<pre>{   "message":{     "channel":"WABA",     "content":{       "preview_url":false,       "type":"TEMPLATE",       "template":{         "templateId":"2003",         "parameterValues":{           "0":"ValueA",           "1":"ValueB"         }       }     }   },   "recipient":{     "to":"919999999999",     "recipient_type":"individual",     "reference":{       "cust_ref":"Some Customer Ref",       "messageTag1":"Message Tag Val1",       "conversationId":"Some Optional Conversation ID"     }   },   "sender":{     "from":"919900000000"   },   "preferences":{     "webHookDNId":"1001"   },   "metaData":{     "version":"v1.0.9"   } }</pre>

### Sample request for Media template

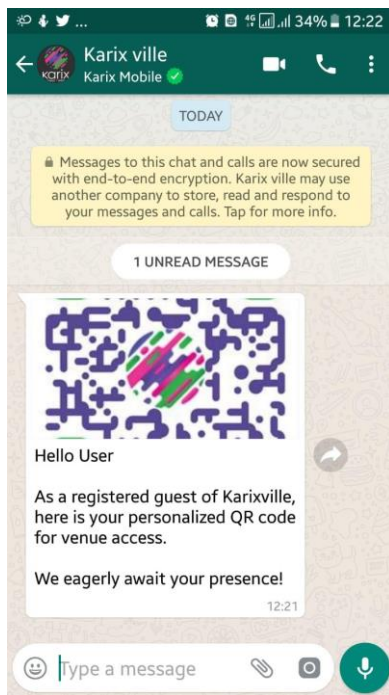
Media template allows you to send content beyond standard Template such as images, Documents & Videos.

The following sample code is an example of sending a Media template of Type Image to WhatsApp

API URL	<code>https://rcmapi.instaalerts.zone/services/rcm/sendMessage</code>
Request Headers	<b>Content-Type:</b> <code>application/json</code> <b>Authentication:</b> <code>Bearer &lt;Account key&gt;</code>
Request Payload	<pre>{   "message": {     "channel": "WABA",     "content": {       "preview_url": false,       "type": "MEDIA_TEMPLATE",       "mediaTemplate": {         "templateId": "kairx_ville_qr",         "media": {           "type": "image",           "url": "https://karix.s3.ap-south- 1.amazonaws.com/QR_Karixville.png"         },         "bodyParameterValues": {           "0": "Sam"         }       }     },     "recipient": {       "to": "919999999999",       "recipient_type": "individual"     },     "sender": {       "from": "919900000000"     },     "preferences": {       "webHookDNId": "1001"     }   },   "metaData": {     "version": "v1.0.9"   } }</pre>



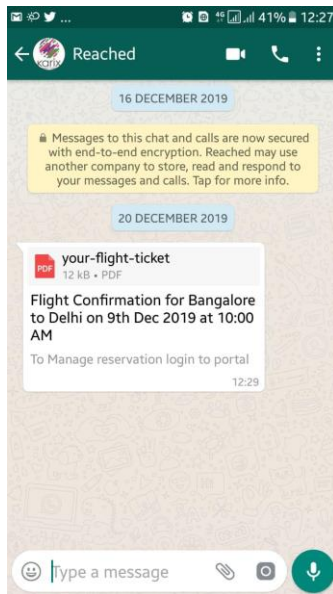
The Media template of Type 'Image' sample is as shown in the below image.



The following sample code is an example of sending a Media template of Type Document to WhatsApp

API URL	<code>https://rcmapi.instaalerts.zone/services/rcm/sendMessage</code>
Request Headers	<b>Content-Type:</b> <code>application/json</code> <b>Authentication:</b> <code>Bearer &lt;Account key&gt;</code>
Request Payload	<pre>{   "message": {     "channel": "WABA",     "content": {       "preview_url": false,       "type": "MEDIA_TEMPLATE",       "mediaTemplate": {         "templateId": "flight_ticket",         "media": {           "type": "document",           "url": "https://www.antennahouse.com/XSLsample/pdf/sample-link_1.pdf",           "fileName": "your-flight-ticket.pdf"         },         "bodyParameterValues": {           "0": "Bangalore to Delhi",           "1": "9th Dec 2019 at 10:00 AM"         }       }     },     "recipient": {       "to": "919999999999",       "recipient_type": "individual"     },     "sender": {       "from": "919900000000"     },     "preferences": {       "webHookDNId": "1001"     }   },   "metaData": {     "version": "v1.0.9"   } }</pre>

The Media template of Type 'Document' sample is as shown in the below image.



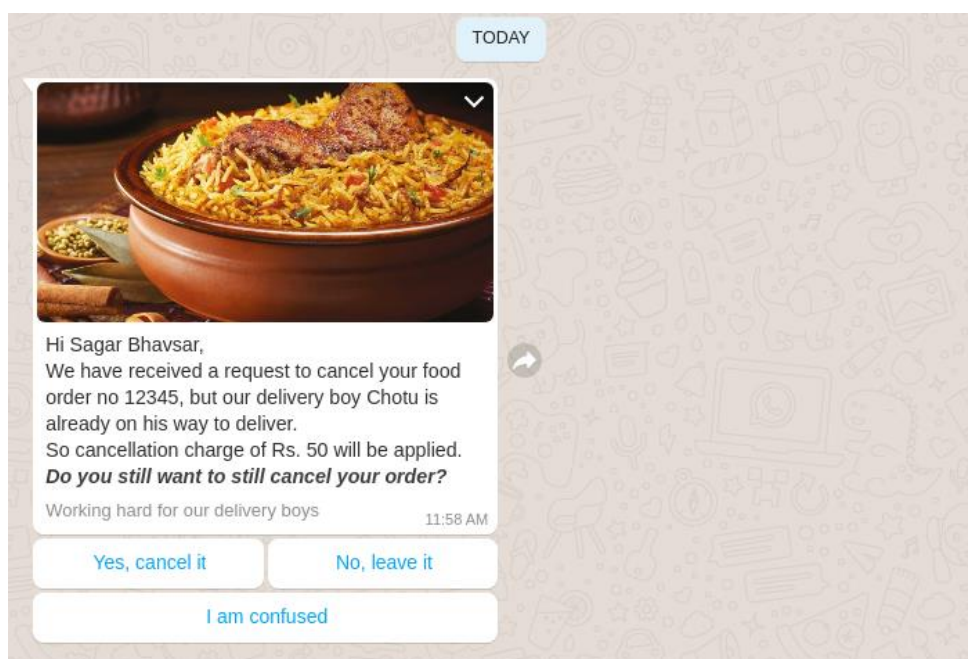
Sample request for quick reply button

The following code provide an example for sending quick reply button:

API URL	<a href="https://rcmapi.instaalerts.zone/services/rcm/sendMessage">https://rcmapi.instaalerts.zone/services/rcm/sendMessage</a>
Request Headers	<b>Content-Type:</b> application/json <b>Authentication:</b> Bearer <Account key>
Request Payload	<pre>{   "message":{     "channel":"WABA",     "content":{       "preview_url":false,       "type":"MEDIA_TEMPLATE",       "mediaTemplate":{         "templateId":"buttons_cancel_food_order",         "media":{           "type":"image",           "url":"https://miro.medium.com/max/780/1*9Wdo1PuiJTZo0Du2A9JLQQ.jpeg"         },         "bodyParameterValues":{           "0":"John",           "1":"12345",           "2":"Chotu",           "3":"50"         },         "buttons":{           "quickReplies":[             {               "index":"0",</pre>

	<pre>       "payload": "{ 'orderid' : '12345', 'reply': 'yes' }"     },     {       "index": "1",       "payload": "e29yZGVyaWQgOiAxMjM0NSwgcmVwbHk6IG5vfQo="     },     {       "index": "2",       "payload": "any other string..."     }   ] } }, "recipient": {   "to": "910000000000",   "recipient_type": "individual",   "reference": {     "cust_ref": "cust_ref123",     "messageTag1": "Message Tag 001",     "conversationId": "Conv_123"   } }, "sender": {   "from": "919999999999" }, "preferences": {   "webHookDNId": "1001" } }, "metaData": {   "version": "v1.0.9" } } </pre>
--	--

The Media template of Type 'Quick-reply button' sample is as shown in the below image.



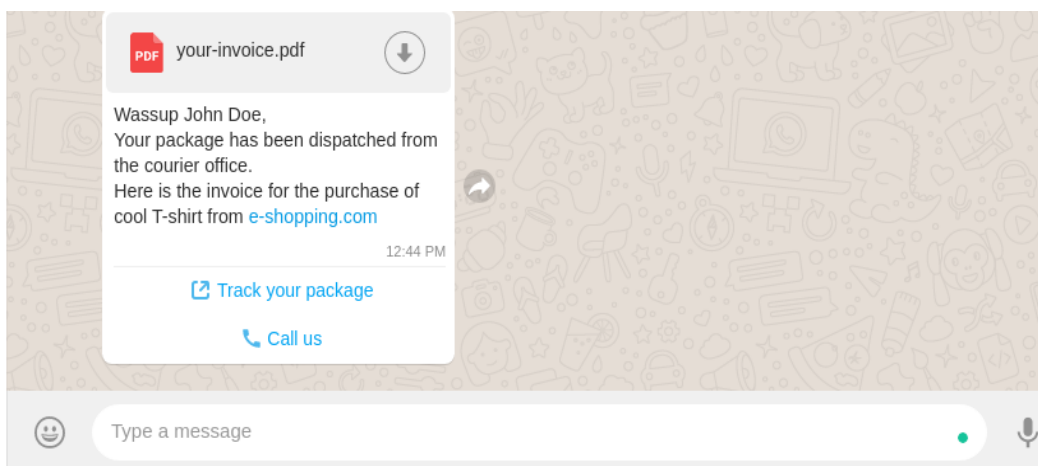
Sample request for call to action

The following code provide an example for sending call to action:

API URL	<a href="https://rcmapi.instaalerts.zone/services/rcm/sendMessage">https://rcmapi.instaalerts.zone/services/rcm/sendMessage</a>
Request Headers	<b>Content-Type:</b> application/json <b>Authentication:</b> Bearer <Account key>
Request Payload	<pre>{   "message":{     "channel":"WABA",     "content":{       "preview_url":true,       "type":"MEDIA_TEMPLATE",       "mediaTemplate":{         "templateId":"buttons_track_me",         "media":{           "type":"document",           "url":"http://www.africau.edu/images/default/sample.pdf",           "fileName":"your-invoice.pdf"         },         "bodyParameterValues":{           "0":"John Bravo",           "1":"cool T-shirt"         },         "buttons":{           "actions":[             {</pre>

	<pre>         "type": "url",         "index": "0",         "payload": "trackingid112233"       }     ]   } }, "recipient": {   "to": "919999999999",   "recipient_type": "individual",   "reference": {     "cust_ref": "cust_ref123",     "messageTag1": "Message Tag 001",     "conversationId": "Conv_123"   } }, "sender": {   "from": "919900000000" }, "preferences": {   "webHookDNId": "1001" } }, "metaData": {   "version": "v1.0.9" } } </pre>
--	---

The Media template of Type 'call-to-actions' sample is as shown in the below image.



### Sample request for video

The following sample code is an example of sending a Media template of Type Video to WhatsApp:

API URL	<code>https://rcmapi.instaalerts.zone/services/rcm/sendMessage</code>
Request Headers	<b>Content-Type:</b> <code>application/json</code> <b>Authentication:</b> <code>Bearer &lt;Account key&gt;</code>
Request Payload	<pre>{   "message":{     "channel":"WABA",     "content":{       "preview_url":true,       "type":"MEDIA_TEMPLATE",       "mediaTemplate":{         "templateId":"test_video",         "media":{           "type":"video",           "url":"http://www.onirikal.com/videos/mp4/assembly_line.mp4"         },         "bodyParameterValues":{           "0":"Life of pi"         }       }     }   },   "recipient":{     "to":"919999999999",     "recipient_type":"individual",     "reference":{       "cust_ref":"cust_ref123",       "messageTag1":"Message Tag 001",       "conversationId":"Conv_123"     }   },   "sender":{     "from":"919900000090"   },   "preferences":{     "webHookDNId":"1001"   },   "metaData":{     "version":"v1.0.9"   } }</pre>

### Sample request for Media message

The following code provide an example for sending Media message of type document

API URL	<code>https://rcmapi.instaalerts.zone/services/rcm/sendMessage</code>
Request Headers	<b><i>Content-Type: application/json</i></b> <b><i>Authentication: Bearer &lt;Account key&gt;</i></b>
Request Payload	<pre>{   "message":{     "channel":"WABA",     "content":{       "type":"ATTACHMENT",       "attachment":{         "type":"document",         "caption":"Some Attachment Text (Caption)",         "mimeType":"application/pdf",         "attachmentData":"JVBERi0xLjQKJ&lt;&lt;attachment DATA base64&gt;&gt;"       }     },     "recipient":{       "to":"919999999999",       "recipient_type":"individual",       "reference":{         "cust_ref":"Some Customer Ref",         "messageTag1":"Message Tag Val1",         "conversationId":"Some Optional Conversation ID"       }     },     "sender":{       "from":"919900000090"     },     "preferences":{       "webHookDNId":"1001"     }   },   "metaData":{     "version":"v1.0.9"   } }</pre>

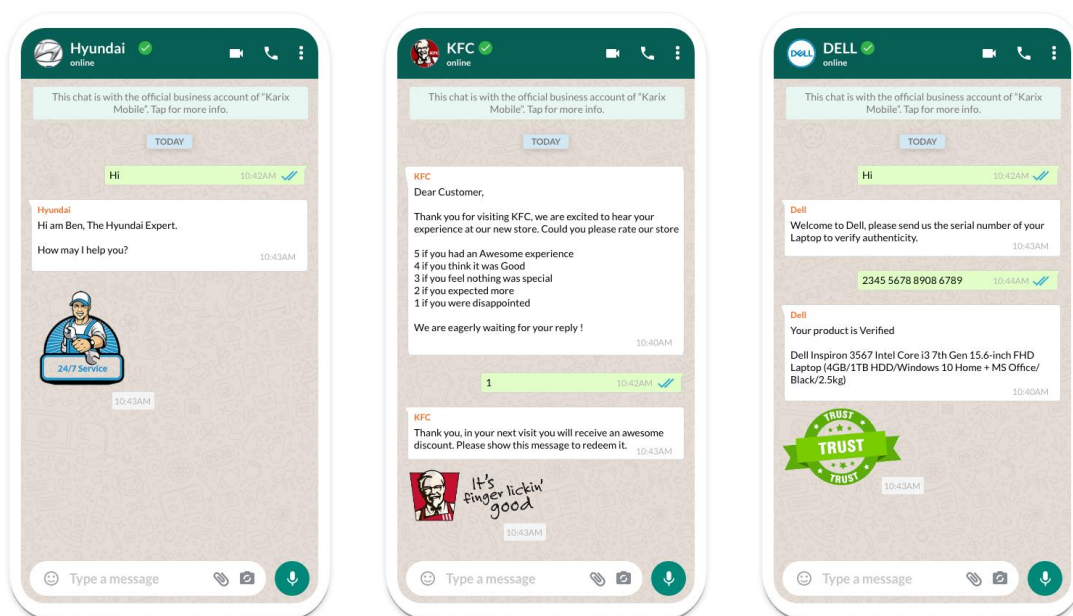


## Sample request for Media message (stickers)

The following code provide an example for sending stickers:

API URL	<code>https://rcmapi.instaalerts.zone/services/rcm/sendMessage</code>
Request Headers	<b>Content-Type:</b> <code>application/json</code> <b>Authentication:</b> <code>Bearer &lt;Account key&gt;</code>
Request Payload	<pre>{   "message":{     "channel":"WABA",     "content":{       "preview_url":false,       "shorten_url":false,       "type":"ATTACHMENT",       "attachment":{         "type":"sticker",  "url":"https://miro.medium.com/max/780/1*9Wdo1PuiJTZo0Du2A9JLQQ.webp"        }     },     "recipient":{       "to":"919999999999",       "recipient_type":"individual",       "reference":{         "cust_ref":"cust_ref123",         "messageTag1":"Message Tag 001",         "conversationId":"Conv_123"       }     },     "sender":{       "name":"Whatsappdemo",       "from":"919900000090"     },     "preferences":{       "webHookDNId":"1001"     }   },   "metaData":{     "version":"v1.0.9"   } }</pre>

The Media message of Type 'Stickers' sample is as shown in the below image.



## Supported Mime Types

Attachment Type	Supported Mime-Types
document	application/pdf, application/msword, application/vnd.ms-powerpoint, application/vnd.ms-excel, text/plain
image	image/jpeg, image/png
audio	audio/acc, audio/mp4, audio/amr, audio/mpeg, audio/ogg, codecs=opus

The maximum size limit of a Media message (Image/Audio/Video) is 16MB.

### Sample request for sending Location

The following code provide an example for sending Location message:

API URL	<code>https://rcmapi.instaalerts.zone/services/rcm/sendMessage</code>
Request Headers	<b><i>Content-Type: application/json</i></b> <b><i>Authentication: Bearer &lt;Account key&gt;</i></b>
Request Payload	<pre>{   "message":{     "channel":"WABA",     "content":{       "preview_url":false,       "type":"LOCATION",       "location":{         "longitude":77.6109,         "latitude":12.9379,         "name":"KARIX MOBILE PRIVATE LIMITED",         "address":" 30, Hosur Main Road"       }     },     "recipient":{       "to":"919999999999",       "recipient_type":"individual"     },     "sender":{       "from":"919900000090"     },     "preferences":{       "webHookDNid":"1001"     }   },   "metaData":{     "version":"v1.0.9"   } }</pre>

### Sample response format

A successful response includes an object with status code, description & mid

<pre>{   "statusCode": "200"   ,"statusDesc": "Successfully Accepted",   "mid": "3111105542939303000" }</pre>
---

## Status Code & Descriptions

The following API status and errors codes that you may get in case there is a wrong parameter entered or issue with the account:

Status Code	Status Description
100	Internal System Error
101	No Service Access
102	Your account is not active
103	Media Not Assigned
104	Credits Expired
105	No credits available in the account
200	Successfully Accepted
201	No Message/ alert message available for the send API
202	Invalid Recipient type: should be individual or group only
203	Invalid Message Type provided
204	No Message Available for the Message
205	Text not Provided
206	Invalid Alert Message Type
207	Invalid Message Type
208	Invalid Quick Reply Missing Text or Payload or Image URL
209	Invalid Quick Reply: Not a valid content type
210	Invalid HSM Data
211	Media to be uploaded
212	Invalid Media to upload-Data Not Provided
213	Invalid Media to upload-Channel Not Found
214	Invalid Media to upload-Channel Does not support given content type
215	Invalid Media to upload Invalid Content Type
216	Invalid Media upload Request
217	Media Id refers to Invalid Content
218	Invalid Channel
219	Source IP Restricted
246	In Buttons, do not use 'Quick replies' and 'Actions' together, use either of them.
429	Too many requests

## Fallback SMS

The Karix RCM API enables support for SMS fallback based on below conditions.

Fallback conditions:

Condition	Default	Description
WhatsApp	Enabled	Fallback to SMS when the targeted MSISDN is not registered in WhatsApp
Number not in Opt-in	Enabled	Fallback to SMS when the targeted MSISDN is not opted for the service.

**Note:** To avail Fallback SMS service contact Karix Accounts manager

### Sample Payload

Fallback SMS Payload	<pre>{   "message": {     "channel": "WABA",     "content": {       "preview_url": false,       "type": "TEMPLATE",       "template": {         "templateId": "availservices",         "parameterValues": {           "0": "Test value 1",           "1": "Testvalue2"         }       }     }   },   "recipient": {     "to": "919999999999",     "recipient_type": "individual"   },   "sender": {     "from": "919900000090"   },   "preferences": {     "webHookDNId": "1001"   },   "smsFallback": {     "sender": "NOTIFY",     "destination": "919999999999",     "message": "This is a SMS fall back msg."   } },   "metaData": {     "version": "v1.0.9"   } }</pre>
----------------------	---

## Webhooks

Webhooks are user-defined HTTP call-backs that are triggered by specific events. Whenever that trigger event occurs, the Call-back engine immediately sends a notification (HTTP request) to the Webhook URL configured in the account updating the status of sent messages or indicating when you receive a message.

**Note:** When a customer sends you a message, the Karix Call-Back service will send an HTTPS POST request notification to the Webhook URL with the details that are described in the User initiated events section.

It is important that your Webhook returns an HTTPS 200 OK response to notifications. Otherwise the Karix Call-back service will consider that notification as failed and do a retrial up to 3 attempts within time delay of 30 secs between each request.

### Types of WhatsApp webhook events Supported by RCM API

There are 2 webhook events

- 1) Delivery events
- 2) User initiated events

#### Delivery events

Delivery events are the status updates of the Message sent to WhatsApp channel, such as success, failed, delivered to the user, read, or deleted etc.

### Sample Payload

Webhook request payload for Delivery Events	<pre>{   "channel": "WABA",   "appDetails": {     "type": "LIVE",     "id": ""   },   "recipient": {     "to": "919999999999",     "recipient_type": "individual",     "reference": {       "cust_ref": "cust_ref123",       "messageTag1": "Message Tag 001",       "conversationId": "Conv_123"     }   },   "sender": {     "from": "919900000090"   },   "events": {     "eventType": "DELIVERY EVENTS",     "timestamp": "1561975547696",     "date": "2019-7-01",     "mid": "3110115354659265000"   },   "notificationAttributes": {     "status": "sent"   "delivered"   "Read"       "reason": "Submitted to channel",   "Delivered to user",   "read by the user"     "code": "100"   } }</pre>
---	---

**Note:** The above sample shows multiple different objects such as audio, document, image, and video for illustration purposes only. A valid request body contains only one of them.

### Sample Payload

The following is the sample webhook request for Delivery Events with **SMS fallback Status**

Webhook request payload for Delivery Events with <b>SMS fallback Status</b>	<pre>{   "channel": "WABA",   "appDetails": {     "type": "LIVE"   },   "recipient": {     "to": "919861439809",     "recipient_type": "individual"   },   "sender": {     "from": "918067329613"   },   "events": {     "eventType": "DELIVERY EVENTS",     "timestamp": "1590683330545",     "date": "2020-5-28",     "mid": "4100527163949897924"   },   "notificationAttributes": {     "status": "Not Sent",     "reason": "Invalid Recipient",     "code": "702"   },   "smsFallback": {     "reason": "Contact is invalid for this service/channel",     "status": "success"   } }</pre>
---	---



## Error code & descriptions

Error Code	Error Description	Error Category	Description
90	Queued	Queued	Queued for Delivery
100	Submitted to channel	Sent	Sent
101	Delivered to User	Delivered	Delivered
102	Read by the user	Read	Read
600	Media not assigned	Dropped	Invalid Permission - RCM Media not enabled for the Account
601	Service not assigned	Dropped	Invalid Permission - WhatsApp service not assigned for the account
602	Number not in Opt-in	Dropped	Number Not in Opt-In
603	Number in Optout	Dropped	Number in Opt-out
604	Number Invalid	Dropped	Invalid Request
605	\ Sender\" Not Found"	Dropped	Invalid Request - WABA Sender Not found
606	Message Request Invalid	Dropped	Invalid Request
607	Template Not Found	Dropped	Template Not Found
608	Submission Retry Exhausted	Dropped	Submission Retry Exhausted
700	Account not ready/restricted	Not Sent	Account related failures - (Messaging For request (On Behalf Of (OBO)) is pending or declined)
701	Account Blocked	Not Sent	Account related failures - Spam rate limit
702	Invalid Recipient	Not Sent	Invalid Recipient
703	User Blocked/Not Authorised	Not Sent	User related Failures
704	Channel Rate Limit Exceeded	Not Sent	System Error - Rate limit hit
705	System Error	Not Sent	System Error while dispatching the request to WhatsApp
706	Error in Send	Not Sent	System Error
707	Error in Message request	Not Sent	Message Error - (Required Parameter Missing)
708	Message type not permitted	Not Sent	Message Error - (Re-engagement message)
801	Message validity Expired	Not Delivered	Message Expired
990	Others	Others	Others

## User initiated events

User initiated message events are the messages that user has sent to the brand such as text, image, or document etc.

### Sample Payloads

Webhook request payload for the user type : <b>Text</b>	<pre>{   "channel": "WABA",   "appDetails": {     "type": "LIVE"   },   "events": {     "eventType": "User initiated",     "timestamp": "1573480550",     "date": "2019-11-11"   },   "eventContent": {     "message": {       "from": "919999999999",       "id": "ABEGkZlABkZgAhAlfMArb76UjC9nUvwOqT8",       "text": {         "body": "Friends"       },       "to": "919900000090",       "contentType": "text"     }   } }</pre>
Webhook request payload for the user reply type : <b>Document</b>	<pre>{   "channel": "WABA",   "appDetails": {     "type": "LIVE"   },   "events": {     "eventType": "User initiated",     "timestamp": "1573481158",     "date": "2019-11-11"   },   "eventContent": {     "message": {       "from": "919999999999",       "id": "ABEGkZlABkZgAhC1kOcdBoWtkHIG9ODTkPfi",       "document": {         "caption": "Fare-phase1",         "filename": "Fare-phase1.pdf",         "mime_type": "application/pdf",         "sha256": "52342221fa3b930f5d9f3ce4fb81c12c76d83cc924c15571ce26abc8b8917584",         "fileLink": "https://rcmmmedia.instaalerts.zone/services/media/download?filedata=dw26ip smOFOWxDHJT8yuNQ"       },       "to": "919900000090",       "contentType": "ATTACHMENT",       "attachmentType": "document"     }   } }</pre>

Webhook request payload for the user reply type : <b>Image</b>	<pre> {   "channel": "WABA",   "appDetails": {     "type": "LIVE"   },   "events": {     "eventType": "User initiated",     "timestamp": "1573480685",     "date": "2019-11-11"   },   "eventContent": {     "message": {       "from": "919999999999",       "id": "ABEGkZlABkZgAhA65twynNyrK-xJDguDFcUp",       "image": {         "mime_type": "image/jpeg", </pre>
--	--

Webhook request payload for the user reply type : <b>Image with caption</b>	<pre> {   "channel": "WABA",   "appDetails": {     "type": "LIVE"   },   "events": {     "eventType": "User initiated",     "timestamp": "1573481095",     "date": "2019-11-11"   },   "eventContent": {     "message": {       "from": "919999999999",       "id": "ABEGkZlABkZgAhAmlv5zJ25c6P_UfAMZ72lM",       "image": {         "caption": "Testing",         "mime_type": "image/jpeg", </pre> <p>"sha256": "59c099f2b7cfae23f8470588f699141768290d4ebbf87ba2d47f1388057db68f",</p> <pre>       "fileLink": "https://rcmmmedia.instaalerts.zone/services/media/download?filedata=dw26ip smOFOWxDHJT8yuNRUO6%2F8UozmFGPd"     },     "to": "919900000090",     "contentType": "ATTACHMENT",     "attachmentType": "image"   } } </pre>
Webhook request payload for the user reply type : <b>Location</b>	<pre> {   "channel": "WABA",   "appDetails": {     "type": "LIVE"   },   "events": {     "eventType": "User initiated",     "timestamp": "1573481446",     "date": "2019-11-11"   },   "eventContent": {     "message": {       "from": "919999999999",       "id": "ABEGkZlABkZgAhALZyShVI4hrmYkt8rMeuRL",       "location": {         "latitude": "13.018301",         "longitude": "80.1873321"       },       "to": "919900000090",       "contentType": "location"     }   } } </pre>

Webhook request  
payload for the  
user reply type :  
**Contacts**

```
{
  "channel": "WABA",
  "appDetails": {
    "type": "LIVE"
  },
  "events": {
    "eventType": "User initiated",
    "timestamp": "1573481521",
    "date": "2019-11-11"
  },
  "eventContent": {
    "message": {
      "from": "919999999999",
      "id": "ABEGkZIABkZgAhCZQLISIW3uSV3emgYx_89v",
      "to": "919900000090",
      "contacts": [
        {
          "addresses": [

          ],
          "emails": [

          ],
          "name": {
            "first_name": "John",
            "formatted_name": "Smith"
          },
          "org": {

          },
          "phones": [
            {
              "phone": "9999999999",
              "type": "Other"
            }
          ],
          "urls": [

          ]
        }
      ],
      "contentType": "contacts"
    }
  }
}
```

Webhook request payload for the user reply type : <b>Video</b>	<pre> {   "channel": "WABA",   "appDetails": {     "type": "LIVE"   },   "events": {     "eventType": "User initiated",     "timestamp": "1573481402",     "date": "2019-11-11"   },   "eventContent": {     "message": {       "from": "919999999999",       "id": "ABEGkZIABkZgAhBfV5_gsWo3vDvkkgKjnbHh",       "video": {         "mime_type": "video/mp4",         "sha256": "7eb2bf1b3556105f1f352f0cac291c48b48ee2afaff8b261e8465761e27e0b27",         "fileLink": "https://rcmmmedia.instaalerts.zone/services/media/download?filedata=dw26ip smOFOWxDHJT8yuNbRIwOzm1wY8bppR15E"       }     },     "to": "919900000090",     "contentType": "ATTACHMENT",     "attachmentType": "video"   } } </pre>
Webhook request payload for the user reply type : <b>Context reply (User replies to a sent message)</b>	<pre> {   "channel": "WABA",   "appDetails": {     "type": "LIVE"   },   "events": {     "eventType": "User initiated",     "timestamp": "1573481585",     "date": "2019-11-11"   },   "eventContent": {     "message": {       "from": "919999999999",       "id": "ABEGkZIABkZgAhAaM6qm1g-gq6lj2EtNr77r",       "text": {         "body": "Thanks so much"       }     },     "to": "919900000090",     "contextmid": "3111119420278357000",     "contentType": "text"   } } </pre>

Webhook request payload for the user reply type : <b>quick-reply button</b>	<pre> {   "channel": "WABA",   "appDetails": {     "type": "LIVE"   },   "events": {     "eventType": "User initiated",     "timestamp": "1573480550",     "date": "2019-11-11"   },   "eventContent": {     "message": {       "from": "919999999999",       "id": "ABEGkZIABkZgAhAlIfMArb76UjC9nUvwOqT8",       "button": {         "payload": "{ 'orderid': '12345', 'reply': 'yes' }",         "text": "Yes, cancel it"       },       "to": "919900000090",       "contentType": "button",       "contextmid": "3111119420278357000"     }   } } </pre>
---	--

#### FileLink

The File link parameter in user initiated events specify the Absolute location of the ContentType:ATTACHMENT ( Image|Document|audio|voice)

To download binary stream of the ATTACHMENT message, make an API call with Header value as below:

Authentication: Bearer (Authentication key)

Note: You can use the same Authentication Key used for post </services/rcm/sendMessage>

## Guidelines

Please make sure to phase your rollout to external users so you can identify and resolve technical issues as they arise. The rollout strategy below is not required but recommended.

We strongly recommend that you start sending and receiving messages with internal employees first to check that the system is working and to prevent negative experiences for your customers.

Once this is done, roll out to 1-5% of your users and make sure that any technical issues are fixed. From there, you can periodically increase the rollout to 10%, 25%, 50%, 75%, and then 100% of your user base. This phased rollout can help you understand and iterate to provide the best user experience.