

WhatsApp Now API Documentation

Table of Contents:

Introduction.....	3
Prerequisites for using WhatsApp Now.....	3
Help and Support.....	3
Understanding WhatsApp Messaging.....	3
Rules for Messaging Customers.....	3
WhatsApp Conversation-Based Pricing.....	4
WhatsApp Now.....	5
Use Case Scenarios.....	5
Generate Token.....	6
Create Template.....	8
Template without Media.....	9
Template with Media.....	10
Advisory for creating Template.....	13
Error Codes for Create Template.....	13
Check Template Status.....	14
Error Codes for Get Template Status.....	16
Send Broadcast.....	17
Error Codes for Send Broadcast.....	18
Send Chat Messages.....	19
Error Codes for Chat Messages.....	20
Setup Webhook for Inbound Messages.....	21
Notification Payload Object.....	21
Value Object.....	23
Message Object.....	25
Statuses Object.....	28

Introduction

Welcome to Message Central's product WhatsApp Now API documentation. Here you will find details to learn about, try, and implement each of the use-cases of WhatsApp Business API platform. WhatsApp Now is a comprehensive solution for integrating with the official WhatsApp Business APIs. Our product leverages the robust and widely-used WhatsApp platform to enable businesses to communicate seamlessly with their customers, providing a reliable and efficient messaging service.

Prerequisites for using WhatsApp Now

Before you begin with API integration, ensure you have:

1. Verified Facebook Business Manager
2. Account on Message Central
3. Linkage of your Facebook business account with Message Central
4. A phone number that needs to be verified and used with Message Central

Help and Support

For implementation support and any feedback, please reach out to us at operations@messagecentral.com

Understanding WhatsApp Messaging

WhatsApp Business Solution messages can only be sent by businesses that have been approved by Meta. This business profile will also have a green verified label to indicate that it is a legitimate business.

The advantage of WhatsApp is that the identifier of users on the platform is their mobile phone number.

Rules for Messaging Customers

WhatsApp has a concept of a 24 hour customer care window, during which a business can freely message an end user. The 24 hour window can be initiated in two ways:

1. An end user sends a message to the business
2. A business sends a templated message to the end user. The 24 hour window starts as soon as the end user replies.

Templates must be approved by Meta before they can be used to send messages to an end user.

When the 24 hour window expires, a new 24 hour customer care window must be initiated again. It is important to be aware that 24 hour customer care windows are not the same as the 24 hour billable conversation window.

WhatsApp Conversation-Based Pricing

Message Central offers a Conversation-Based Pricing model for WhatsApp messaging. This is in line with the pricing model introduced by WhatsApp at the start of February 2022.

What is a conversation?

A conversation is any number of messages sent within a 'session', which is defined as a 24 hour period starting from the time the first message is sent by the business.

Who initiates a conversation?

A conversation can be *initiated* either by a *customer* (user-initiated) or *business* (business-initiated), but in either case the 'session' begins with the first message sent by the business.

How does the pricing per conversations work?

The pricing for WhatsApp messaging is priced per conversation. Starting June 1st, this is priced per conversation and conversation type. There are now four conversation categories, each of which are [charged at different rates](#):

1. **Utility Conversation** - Enable the completion of a particular, agreed-upon request or transaction or provide a customer with an update about an ongoing transaction, including post-purchase notifications and regular billing statements.
2. **Authentication Conversation** - Enables businesses to utilize one-time passcodes to verify users at various login stages, if necessary (e.g. account verification, account recovery, integrity challenges)
3. **Marketing Conversation** - Business-initiated conversations to market a good or service to customers, such sending pertinent offers to customers who have opted in. This would also include any business-related interaction that is not an authentication or utility conversation.
4. **Service Conversation** - User-initiated discussions that assist consumers in getting answers to their questions.

NOTE: If a conversation contains more than one category of message, you will be charged for both conversation types:

WhatsAppNow APIs

Message Central facilitates the sending of messages over various communication channels, including SMS, WhatsApp, RCS (Rich Communication Services). This API provides robust features and flexible options to meet diverse messaging needs. WhatsApp Now is a product of Message Central which is based on official WhatsApp Business APIs.

Use Case Scenarios

- **Promotional Campaigns:** Use WhatsApp Now to send promotional as well as utility messages to customers via WhatsApp. Send multimedia and documents directly through WhatsApp. Deliver personalized messages, notifications, and updates to enhance customer satisfaction and loyalty to enhance your customer experience.
- **Live-Chat Messages:** Send and receive messages, multimedia, and documents directly through WhatsApp, ensuring timely and effective communication with your audience.
- **OTP Verification:** Implement a secure OTP delivery system for user authentication, choosing between SMS, WhatsApp, or email based on user preferences and security requirements.

Also, the following parameters need to be sent while using WhatsAppNow APIs;

API Parameter	Type	Value
flowType	String	SMS/WHATSAPP/RCS/SAUTH
messageType	String	OTP

NOTE: If you want to send OTP messages via WhatsApp, Message Central recommends using Verify Now APIs, which is specifically designed for that use case.

Rest API Base URLs:

All Platform API endpoints below should be prefixed with the following URL:

<https://cpaas.messagecentral1.com>

Generate Token

When using the WhatsApp Now APIs to create templates, send text messages and broadcast messages, the initial call should be to the token generation API. This API returns a token that must be included in all subsequent calls. An authentication token is needed to validate the user and should be included in the header section of each request.

Request Parameters:

Field	Type	Mandatory	Description
customerId	String	yes	Customer identifier (need to login on message central website to get your customer id)
country	Integer	no	Country code to send OTP to
email	String	no	Email
key	String	yes	Base-64 encrypted password
scope	String	no	Use 'NEW' for first time

Request URL Path:

```
/auth/v1/authentication/token
```

cURL

```
curl --location  
'https://cpaas.messagecentral.com/auth/v1/authentication/token?customerId=<  
CustomerId>&key=<Base64 Encrypted  
password>&scope=NEW&country=91&email=test@messagecentral.com' \  
--header 'accept: */*'
```

NOTE: To convert a cURL command into code using Postman, open Postman, import the cURL command via the "Import" button, and then generate the code in your preferred language by clicking the "Code" button on the right side of the request.

Response JSON:

A successful response will return a **200** status code.

```
{  
  "status": Integer,  
  "token": "String"  
}
```

Create Template

You can create templates of various types and categories, as well as creating multiple language variations of a template. When creating templates with multiple language versions, make sure to be consistent with translations across the versions.

Make sure that your templates follow [WhatsApp Message Template Guidelines](#). Not correctly following the guidelines can impact on templates being approved.

Approval Process

Templates need to be approved by Meta before they can be used in a WhatsApp message. When created, templates initially have a status of **PENDING**. Once approved, they will have a status of **APPROVED**, and can then be used.

Template Categories

There are now three supported template categories for each of the business-initiated conversation types:

- **UTILITY**: Enable the completion of a particular, agreed-upon request or transaction or provide a customer with an update about an ongoing transaction, including post-purchase notifications and regular billing statements.
- **AUTHENTICATION**: Enables businesses to utilize one-time passcodes to verify users at various login stages, if necessary (e.g. account verification, account recovery, integrity challenges).
- **MARKETING**: Business-initiated conversations to market a good or service to customers, such sending pertinent offers to customers who have opted in. This would also include any business-related interaction that is not an authentication or utility conversation.

To **createTemplate** below are the request parameters. The authentication token is required to create a template which is generated by the generated token API (which you can find above in Introduction section)

Request Header	Type	Mandatory
authToken	String	yes

Request URL Path:

POST /verification/v3/template

Template without Media

This allows you to design and manage message templates that consist solely of text, without any multimedia content. These templates can be used for various communication purposes, such as sending notifications, alerts, or reminders to your customers through WhatsApp.

Request Parameters:

Field	Type	Mandatory	Description
phoneNumber	String	yes	Your WABA number (eg - 917714836906)
name	String	yes	Name of your template.
language	String	yes	The language of the template. A list of supported languages is available in the WhatsApp documentation . (Eg: en_US)
category	String	yes	The required category of the template. Possible values can include MARKETING , UTILITY , etc.
headerFormat	String	yes	The format of the template component. It can be TEXT
header	String	yes	The header text of the message, that appears at the top of the message.
body	String	yes	The main content of the message template. It can also contain variables.
footer	String	no	The footer text of your message content.
bodyValue	String	no	These are example value for your variable template which will be required by Meta for your template approval in case of Variable template (eg: Hello {{1}} from Message Central..... bodyValue: John)

buttons	String	no	<p>An array of objects representing button components.</p> <p>You need to provide with QUICK_REPLY, URL(Static or Dynamic), PHONE_NUMBER etc.</p> <p>Eg - [{"type":"PHONE_NUMBER", "text":"Call Us", "value":"918750844671"}, {"type":"URL", "text":"Visit Us", "value":"https://www.example.com/my_url"}]</p>
---------	--------	----	---

Payloads for different CTA buttons:

These buttons can be attached to text messages or media messages. Once your interactive message templates have been created and approved, you can use them in notification messages as well as customer service/care messages.

Please refer to the following constraints before creating a message template.

- For call-to-action templates, you can add 2 buttons, up to one button of each type (call phone number and visit website).
- For quick reply templates, you can add up to 3 buttons.

Static URL case:

```
{
  "type": "URL",
  "text": "Static Link text",
  "url": "https://www.example.com"
}
```

Dynamic URL case:

```
{
  "type": "URL",
  "text": "Dynamic Link text",
  "url": "https://www.example.com/{{1}}",
  "example": "https://www.example.com/test"
}
```

Phone Number case:

```
{
  "type": "PHONE_NUMBER",
  "text": "Call Us",
  "value": "917715234323"
}
```

cURL

```
curl --location 'https://cpaas.messagecentral.com/verification/v3/template' \
--header 'authToken:
eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJDLTM4MERBQzghchgvjevREMSIsIm1hdCI6MTczNzcxN
TkwMSwiZXhwIjoxODk1Mzk1OTAxZQ.3dwSoktsxUvSnkMd405CgeB5Fnuh2AvhVnFS5BeK8FEW8
za2cieWh-K-35HUdkYAuy8YTS1xW0cF007RR5ZRw' \
--form 'phoneNumber="919667884599"' \
--form 'name="template_without_media"' \
--form 'language="en_US"' \
--form 'category="MARKETING"' \
--form 'headerFormat="TEXT"' \
--form 'header="Congratulations"' \
--form 'body="Hello Your account recharge of ₹100 is successful. Please use
our services and enjoy\!"' \
--form 'buttons="[{"type\":"PHONE_NUMBER\","text\":"Contact Us\","
"value\":"917750844671\"},{\"type\":"URL\","text\":"Visit Us\","
"value\":"https://console.messagecentral.com\"}]"' \
--form 'footer="Type \\"STOP\\" to unsubscribe"'
```

Template with Media

This enables you to design and manage message templates that include multimedia content such as images, videos, and documents. These templates are ideal for rich and engaging communications with your end-customers.

Request Parameter:

Field	Type	Mandatory	Description
phoneNumber	String	yes	Your WABA number (eg - 917714836906)
name	String	yes	Name of your template.
language	String	yes	The language of the template. A list of supported languages is available in the WhatsApp documentation . (Eg: en_US)
category	String	yes	The required category of the template. Possible values can include MARKETING , UTILITY , etc.
headerFormat	String	yes	The format of the template component. It must be only , IMAGE , VIDEO or DOCUMENT
file	Multipart File	yes	The media file associated with the template header. Eg: Image file
body	String	yes	The main content of the message template. It can also contain variables.
footer	String	no	The footer text of your message content.
bodyValue	String	no	These are example value for your variable template which will be required by Meta for your template approval in case of Variable template
buttons	String	no	<p>An array of objects representing button components.</p> <p>You need to provide with QUICK_REPLY, URL, PHONE_NUMBER etc.</p> <p>Eg - [{"type":"PHONE_NUMBER", "text":"Call Us", "value":"918750844671"}, {"type":"URL", "text":"Visit Us", "value":"https://www.example.com/my_url"}]</p>

cURL

```
curl --location 'https://cpaas.messagecentral.com/verification/v3/template' \
--header 'authToken:
eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJDLTM4MERBQzg1RDlGRjREMSIsIm1hdCI6MTcyMDc2OT
I2OCwiZXhwIjoxODc4NDQ5MjY4fQ.08x17vMUoOU4dCn61dqFTjBMxTVvnEpRCKzjMo4JW9YQrE
dL0PK3sQW4PgZBWRUCvKDkoSRoDAdXYh9rxa' \
--form 'phoneNumber="919457888189"' \
--form 'name="template_without_media"' \
--form 'language="en_US"' \
--form 'category="MARKETING"' \
--form 'headerFormat="TEXT"' \
--form 'body="Hello Your account recharge of ₹100 is successful. Please use
our services and enjoy!"' \
--form 'footer="Type \'STOP\' to unsubscribe"' \
--form 'file=@/C:/Users/Kunal Suryawanshi/Downloads/Media (9).png'
```

Response JSON (without & with Media Templates):

A successful response will return a **200** status code.

```
{
  "responseCode": 200,
  "data": {
    "data": {
      "name": "example_template",
      "id": "790937216444187",
      "status": "PENDING",
      "category": "MARKETING"
    }
  }
}
```

NOTE: You will get “**template name**” in **response body** after creating a new template. You must use this formatted template name subsequently when checking the template status and creating a broadcast, rather than the name originally used to create the template. **For example:** If the user enters 'Abhishek786', the response will return it as 'abhishek_786'. You will get the template status directly from Meta. It can be APPROVED, PENDING or REJECTED. You will get the correct template status by using check template status api.

Advisory for creating Template

- Name** - The name of the template should always be in small letters. Also, in case of multiple words please add a “_” in between 2 words (eg: tes_template_01)
- Variable template** - To create a variable template, you need to add placeholders and their body values mandatorily in while creating a request for such a template.
Variables - They are always added in two curly braces, and in numeric increasing order, otherwise your template will get rejected.
bodyValue - These are examples of what will go inside the variable values. Since these are form data, hence in case of more than 2 variables, separate them by spaces.
Sample message - Hello {{1}}, get 50% discount by applying the coupon code {{2}} latest by {{3}}
bodyValue(for above message) - “Kunal MSGCTL50 20thJune”, here Kunal is an example of 1st bodyValue and so on.
- Character limit** - A text message can be a max of **4096** characters long.

Error Codes for Create Template

Code	Description
801 Invalid sender	phoneNumber not mentioned OR Incorrect phoneNumber used
810 Required Parameter is missing. Template name, type, category and language are required	name not mentioned OR category not mentioned OR language not mentioned OR header_format not mentioned OR body not mentioned
800 (#100) Param category must be one of {UTILITY, MARKETING, AUTHENTICATION} - got \"MARKETING\".	Incorrect category used
800 Content can't be added for this language because it is not available for message templates. Please try	Incorrect language used

again with a different language.	
811 Invalid header_format	Incorrect header_format
812 File is not allowed in header_format TEXT	You cannot send a file while creating a template without Media
813 Header is mandatory if header_format is TEXT	header_format is missing
814 File is mandatory if header_format is media type	File is missing if your headerFormat is IMAGE, VIDEO or DOCUMENT
815 Header is not allowed if header_format is file type	Header should not be used while creating a Template with media
826 Variable template but missing bodyValues field	bodyValue is mandatory in case of variable template
827 bodyValue does not match with number of placeholders/variables used in body	bodyValue mismatch in case of variable template

Check Template Status

This allows you to monitor the status of your message templates submitted for approval on WhatsApp. This API provides real-time updates on whether your templates are approved, pending, or rejected, ensuring you stay informed about their availability for use.

Request Parameters:

Field	Type	Mandatory	Description
phoneNumber	String	yes	Your WABA number (eg - 917715436906)
templateName	String	yes	Name of your template which you used To create the template
templateId	String	yes	This is the unique id for the created template. You will get this id in response body of your created template
templateLang	String	yes	The language of created template (eg: en_US)

cURL

```
curl --location --request GET
'https://cpaas.messagecentral.com/verification/v3/template' \
--header 'authToken:
eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJDLTM4MERBQzg1RDlGRjREMSIsIm1hdCI6MTcyMDc2OT
I2OCwiZXhwIjoxODc4NDQ5MjY4fQ.08x17vMUoOU4dCn61dqFTjBMxTVvnEpRCkzjMo4JW9YQrE
dL0PK3sQW4PgZBWRUCvKDkoSRoDAdYXh9rxa' \
--form 'phoneNumber="919547888189"' \
--form 'templateId="205681076471259"' \
--form 'templateName="test_template"'
```

Response JSON:

A successful response will return a **200** status code.

```
{
  "name": "prod_test_temp_wmedia_1",
  "components": [
    {
      "type": "HEADER",
      "format": "TEXT",
      "text": "Congratulations!"
    },
    {
      "type": "BODY",
      "text": "Your account recharge of $100 is successful. Please
use our services and enjoy!"
    },
    {
      "type": "FOOTER",
      "text": "Type \"STOP\" to unsubscribe"
    }
  ],
  "language": "en_US",
  "status": "APPROVED",
  "category": "UTILITY",
  "id": "205681076471259"
}
```


Error Codes for Get Template Status

Code	Description
801 Invalid sender	phoneNumber not mentioned
816 Provide a correct template name	Invalid template name
817 Required parameter is missing. Template id and Template name are required	Template id and Template name are missing
800 Unsupported get request. Object with ID '1021907252855200' does not exist, cannot be loaded due to missing permissions, or does not support this operation. Please read the Graph API documentation at https://developers.facebook.com/docs/graph-api	Invalid template id

Send Broadcast

To send a WhatsApp broadcast to mobile numbers, the following request parameters are required. An authentication token, generated by the token generation API, is necessary to send the template messages via broadcast.

Request Header	Type	Mandatory
authToken	String	yes

Request URL Path:

POST	<code>/verification/v3/send</code>
-------------	------------------------------------

Send Single Broadcast

To send a WhatsApp broadcast message to a single recipient for personalized outreach, customer support responses, or urgent notifications where targeted communication is required.

Request Parameters:

Field	Type	Mandatory	Description
flowType	String	yes	It must be "WHATSAPP"
type	String	yes	It must be "BROADCAST"
templateName	String	yes	Your approved template name which you want to send as a broadcast
senderId	String	yes	Your WABA number (eg: 917715836606)
countryCode	String	yes	Country code
mobileNumber	String	yes	Mobile number for single text
langId	String	yes	Lang of created template (eg: en_US)
variables	String	no	These are values inserted into the template body placeholders Eg: "John, MSG3453, 3445" (for multiple placeholders)

ctaVariables	String	no	These are values inserted inside the dynamic CTA placeholders (comma separated) Eg: "234, ver453" (for multiple placeholders)
--------------	--------	----	--

NOTE: The templates used must be APPROVED from Meta. A maximum of 2 dynamic CTAs can only be sent.

cURL

```
curl --location --request POST
'https://cpaas.messagecentral.com/verification/v3/send?flowType=WHATSAPP&type=BROADCAST&mobileNumber=7715836906&countryCode=91&senderId=919667888109&langId=en_US&templateName=demo_template&variables=Kunal%2C%20MSG3545&ctaVariables=234%2C%20ver453' \
--header 'authToken:
eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJDLTM4MERBQzg1RDlGRjREMSIsIm1hdCI6MTczNzcwNTkwMSwiZXhwIjoxODk1Mzk1OTAxZQ.3uyhwSoktsxUvSnkMd405CgeB5Fnuh2AvhVnFS5BeK8FEW8za2cieWh-_K-35HUdkYAuy8YTS1xW0cF05ZRw'
```

Response JSON:

A successful response will return a **200** status code.

```
{
  "responseCode": 200,
  "message": "SUCCESS",
  "data": null
}
```

Send Bulk Broadcast (Using Excel)

To send a WhatsApp broadcast message in bulk by uploading an Excel file containing recipient details. This method is useful when handling dynamic recipient lists, allowing users to personalize messages by including variables such as names or custom fields from the spreadsheet.

Request Parameters:

Field	Type	Mandatory	Description
flowType	String	yes	It must be "WHATSAPP"
type	String	yes	It must be "BROADCAST"
templateName	String	yes	Your approved template name which you want to send as a broadcast
senderId	String	yes	Your WABA number (eg: 917715836606)
file	Multipart file XLSX format	yes	This needs to be send as form data (Please check the format below)
langId	String	yes	Lang of created template (eg: en_US)

NOTE: The template used must all be APPROVED templates from Meta.

Please find the Sample file format in XLXS format below:

Country	Mobile	ctaVar1	ctaVar2	var1	var2	var3

Field Descriptions:

- **Country** – Country code of the recipient.
- **Mobile** – Mobile number for sending a single text.
- **ctaVar1** – Value for the 1st dynamic CTA placeholder. If your template does not include a dynamic CTA, keep this column in the header but leave the values blank.
- **ctaVar2** – Value for the 2nd dynamic CTA placeholder. If your template does not include a second dynamic CTA, keep this column in the header but leave the values blank.
- **var1** – Value for the 1st body placeholder in the template.
- **var2** – Value for the 2nd body placeholder in the template.
- **var(n)** – Value for the nth body placeholder in the template

NOTE: The first four columns (**Country**, **Mobile**, **ctaVar1**, **ctaVar2**) **must always be included in the header**, even if there are no dynamic CTAs in the template. In such cases, the column names should remain, but the values should be left blank.

The number of **variable** columns in the Excel file should match the number of variable placeholders in your template exactly. For example, if your template contains two placeholders, your file must include **var1** and **var2**—no more, no less.

Please refer to the following examples to understand the correct file formatting.

Example 1: If an enterprise wants to send a customized message using file upload with no dynamic CTA and 2 variables, the message field should be formatted as follows:

Sample message - Hello {{1}}, get 50% discount by applying the coupon {{2}}
The XLXs format should be as follows:

Country	Mobile	ctaVar1	ctaVar2	var1	var2
91	7890987652			Kunal	MSG3453

Example 2: If an enterprise wants to send a customized message using file upload with dynamic CTA and 3 variables, the message field should be formatted as follows:

Sample message - Hello {{1}}, get 50% discount by applying the coupon {{2}} latest by {{3}}
CTA - www.messagecentral.com/{{1}}

The XLXs format should be as follows:

Country	Mobile	ctaVar1	ctaVar2	var1	var2	var3
91	7890987652	v123		Kunal	MSG3453	21st Jan

cURL

```
curl --location
'https://cpaas.messagecentral.com/verification/v3/send?flowType=WHATSAPP&senderId=919457848169&type=BROADCAST&templateName=template_without_media_1' \
--header 'authToken:
eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJDLTM4MERBQzg1RDlGRjREMSIsIm1hdCI6MTcyMDc2OTI2MCwiZXhwIjoxODc4NDQ5MjY4fQ.08x17vMUoOU4dCn61dqFTjBMxTVvnEpRCKzjMo4JW9YQrEdL0PK3sQW4PgZBWRUCvKDkoSRoDAdYXh9rxa' \
--form 'file=@"/C:/Users/Kunal Suryawanshi/Documents/Test_WA_API.xlsx''
```

Response JSON:

A successful response will return a **200** status code.

```
{
  "responseCode": 200,
  "message": "SUCCESS",
  "data": null
}
```

Error Codes for Send Broadcast

Code	Description
800 Invalid Auth	Invalid Auth token
801 Invalid sender	senderId is missing or Incorrect senderId used
802 Invalid Template	Template name or language is missing or incorrect
803 Contact list not found.	Contact sheet missing
804 Number of parameters doesn't match	Contact sheet there but variable params not matching
805 insufficient credits	Credit is lower than required to create successful broadcast campaign

Send Chat Messages

To send a WhatsApp message(chat message) to a mobile number the following request parameters are required. An authentication token, generated by the token generation API, is necessary to send any messages over live-chat.

Request Header	Type	Mandatory
authToken	String	yes

Request URL Path:

POST	<code>/verification/v3/send</code>
-------------	------------------------------------

Request Parameters:

Field	Type	Mandatory	Description
flowType	String	yes	It must be "WHATSAPP"
type	String	yes	It must be "CHAT"
senderId	String	yes	Your WABA number (eg - 917715836908)
countryCode	String	yes	The country code of the recipient's mobile number
mobileNumber	String	yes	Recipient's mobile number
message	String	yes	The content of the chat message
file	Multipart File	no	Image, video which you want to send

cURL

```
curl --location --request POST
'https://cpaas.messagecentral.com/verification/v3/send?countryCode=91&flowType=WHATSAPP&mobileNumber=7715836906&senderId=919457888189&type=CHAT&message=Welcome%20to%20Message%20Central' \
--header 'authToken:
eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJDLTM4MERBQzgz1RDlGRjREMSIsIm1hdCI6MTcyMDc2OTI2MCwiZXhwIjoxODc4NDQ5MjY4fQ.08x17vMUoOU4dCn61dqFTjBMxTVvnEpRCKzjMo4JW9YQrEdL0PK3sQW4PgZBWRUCvKDkoSRoDAdYXh9rxa'
```

Response JSON:

A successful response will return a **200** status code.

```
{
  "responseCode": 200,
  "message": "SUCCESS",
  "data": {
    "verificationId": null,
    "mobileNumber": "917715836906",
    "responseCode": "200",
    "errorMessage": null,
    "timeout": null,
    "smsCLI": null,
    "transactionId":
    "wamid.HBGmOTE3NzE1ODM2OTA2FQIAERGSOEFFNUM5MTFDRUVBMDUzMDFGAA=="
  }
}
```

Error Codes for Chat Messages

Code	Description
800 Invalid Auth	Invalid Auth token
801 Invalid sender	senderId is missing OR Incorrect senderId used
805 insufficient credits	Credit is lower than required to send a chat message
829 mobileNumber is required	Recipient's mobile number is missing
830 message is required	The content of chat message is missing

Setup Webhook for WhatsApp Messages

Before you can start receiving notifications you will need to create an endpoint on your server to receive notifications.

Whenever a trigger event occurs, the Message Central's WhatsApp Business Platform sees the event and sends a notification to a Webhook URL you will specify. You can get two types of notifications:

1. **Received messages:** This alert lets you know when you have received a message. These can also be called "inbound notifications" throughout the documentation.
2. **Message status:** This alert lets you know when the status of a message has changed—for example, the message has been read or delivered. These can also be called "outbound notifications."

To subscribe to Webhooks, you will need to follow these steps:

1. **Information Required:**
 - a. Customer ID
 - b. Brand Name
 - c. Registered Email ID
 - d. Callback URL
2. **Submission Instructions:** Please send the above information to operations@messagecentral.com with the subject line "WhatsApp Webhook Configuration with Message Central".
3. **Callback URL Specifications:** Ensure your callback URL is accessible and capable of receiving HTTP POST requests containing JSON payloads with status updates. Upon triggering, the endpoint will receive the following JSON payload as described below in the "Notification Payload Object" and it must respond with 200 .

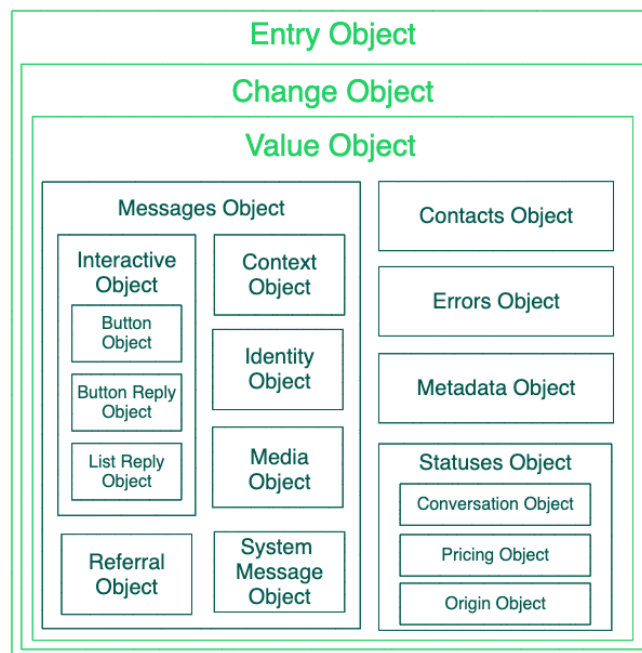
Notification Payload Object

The notification payload is a combination of nested objects of JSON arrays and objects that contain information about a change.

Example Notification Payload

```
{
  "object": "whatsapp_business_account",
  "entry": [{
    "id": "WHATSAPP-BUSINESS-ACCOUNT-ID",
    "changes": [{
      "value": {
        "messaging_product": "whatsapp",
        "metadata": {
          "display_phone_number": "PHONE-NUMBER",
          "phone_number_id": "PHONE-NUMBER-ID"
        },
        # Additional arrays and objects
        "contacts": [{...}]
        "errors": [{...}]
        "messages": [{...}]
        "statuses": [{...}]
      },
      "field": "messages"
    }
  ]
}]
}
```

The Nested Structure of the Notification Payload



Name	Type	Description
object	String	The specific webhook a business is subscribed to. The webhook is <code>whatsapp_business_account</code> .
entry	array of objects	<p>An array of entry objects. Entry objects have the following properties:</p> <ul style="list-style-type: none"> • id - <i>String</i>. The WhatsApp Business Account ID for the business that is subscribed to the webhook. • changes - <i>Array of objects</i>. An array of change objects. Change objects have the following properties: <ul style="list-style-type: none"> ◦ value - <i>Object</i>. A value object. See Value Object. ◦ field - <i>String</i>. Notification type. Value will be <code>messages</code>

Value Object

The value object contains details for the change that triggered the webhook. This object is nested within the changes array of the entry array.

Name	Type	Description
contacts	array of objects	<p>Array of contact objects with information for the customer who sent a message to the business. Contact objects have the following properties:</p> <ul style="list-style-type: none"> • wa_id — <i>String</i>. The customer's WhatsApp ID. A business can respond to a customer using this ID. This ID may not match the customer's phone number, which is returned by the API as input when sending a message to the customer. • profile — <i>Object</i>. A customer profile object. Profile objects have the following properties: <ul style="list-style-type: none"> ◦ name — <i>String</i>. The customer's name.
errors	array of objects	An array of error objects describing the error. Error objects have the following properties, which map to their equivalent properties in API error response payloads.

		<p>Webhooks triggered by v15.0 and older requests:</p> <ul style="list-style-type: none"> • code — Integer. Example: 130429. • title — String. Error code title. Example: Rate limit hit. <p>Webhooks triggered by v16.0 and newer requests:</p> <ul style="list-style-type: none"> • code — Integer. Error code. Example: 130429. • title — String. Error code title. Example: Rate limit hit. • message — String. Error code message. This value is the same as the title value. For example: Rate limit hit. Note that the message property in API error response payloads pre-pends this value with the a # symbol and the error code in parenthesis. For example: (#130429) Rate limit hit. • error_data — Object. An error data object with the following properties: <ul style="list-style-type: none"> ◦ details — String. Describes the error. Example: Message failed to send because there were too many messages sent from this phone number in a short period of time.
messages	array of objects	Information about a message received by the business that is subscribed to the webhook. See Messages Object .
metadata	object	<p>A metadata object describing the business subscribed to the webhook. Metadata objects have the following properties:</p> <ul style="list-style-type: none"> • display_phone_number — String. The phone number that is displayed for a business. • phone_number_id — String. ID for the phone number. A business can respond to a message using this ID.
statuses	array of objects	Status object for a message that was sent by the business that is subscribed to the webhook. See Statuses Object .

Message Object

The messages array of objects is nested within the value object and is triggered when a customer sends a message to the business that is subscribed to the webhook.

Name	Type	Description
button	object	<p>When the messages type field is set to button, this object is included in the messages object:</p> <ul style="list-style-type: none"> payload – String. The payload for a button set up by the business that a customer clicked as part of an interactive message. text – String. Button text.
context	object	<p>Context object. Only included when a user replies or interacts with one of your messages. Context objects can have the following properties:</p> <ul style="list-style-type: none"> from – String. The WhatsApp ID for the customer who replied to an inbound message. id – String. The message ID for the sent message for an inbound reply.
document	String	<p>A document object. When messages type is set to document, this object is included in the messages object. Document objects can have the following properties:</p> <ul style="list-style-type: none"> caption – String. Caption for the document, if provided. filename – String. Name for the file on the sender's device. sha256 – String. SHA 256 hash. mime_type – String. Mime type of the document file. id – String. ID for the document.
errors	array of objects	<p>An array of error objects describing the error. Error objects have the following properties, which map to their equivalent properties in API error response payloads</p> <p>Webhooks triggered by v15.0 and older requests:</p> <ul style="list-style-type: none"> code – Integer. Example: 130429. title – String. Error code title. Example: Rate limit hit.

		<p>Webhooks triggered by v16.0 and newer requests:</p> <ul style="list-style-type: none"> • code — Integer. Error code. Example: 130429. • title — String. Error code title. Example: Rate limit hit. • message — String. Error code message. This value is the same as the title value. For example: Rate limit hit. Note that the message property in API error response payloads pre-pends this value with the a # symbol and the error code in parenthesis. For example: (#130429) Rate limit hit. • error_data — Object. An error data object with the following properties: <ul style="list-style-type: none"> ◦ details — String. Describes the error. Example: Message failed to send because there were too many messages sent from this phone number in a short period of time.
from	String	The customer's WhatsApp ID. A business can respond to a customer using this ID. This ID may not match the customer's phone number, which is returned by the API as input when sending a message to the customer.
id	object	The ID for the message that was received by the business. You could use messages endpoint to mark this specific message as read.
image	object	<p>When messages type is set to image, this object is included in the messages object.</p> <ul style="list-style-type: none"> • caption — String. Caption for the image, if provided. • sha256 — String. Image hash. • id — String. ID for the image. • mime_type — String. Mime type for the image.
sticker	object	<p>When messages type is set to sticker, this object is included in the messages object. Sticker objects have the following properties:</p> <ul style="list-style-type: none"> • mime_type — String. image/webp. • sha256 — String. Hash for the sticker. • id — String. ID for the sticker. • animated — Boolean. Set to true if the sticker is animated; false otherwise.

text	object	<p>When the message type is set to text, this object is included.</p> <p>Text objects have the following properties:</p> <ul style="list-style-type: none"> • body — String. The text of the message.
timestamp	String	<p>Unix timestamp indicating when the WhatsApp server received the message from the customer.</p>
type	String	<p>The type of message that has been received by the business that has subscribed to Webhooks. Possible value can be one of the following:</p> <ul style="list-style-type: none"> • button • text • image • sticker • video
video	object	<p>When messages type is set to video, this object is included in messages object. Video objects have the following properties:</p> <ul style="list-style-type: none"> • caption – String. The caption for the video, if provided. • filename – String. The name for the file on the sender's device. • sha256 – String. The hash for the video. • id – String. The ID for the video. • mime_type – String. The mime type for the video file.

Statuses Object

The statuses object is nested within the value object and is triggered when a message is sent or delivered to a customer or the customer reads the delivered message sent by a business that is subscribed to the Webhooks.

Name	Type	Description
conversation	Object	<p>Information about the conversation.</p> <p>id - Represents the ID of the conversation the given status notification belongs to.</p> <p>origin - Describes conversation category</p> <p>type - Indicates conversation category. This can also be referred to as a conversation entry point</p> <p>authentication - Indicates the conversation was opened by a business sending template categorized as AUTHENTICATION to the customer.</p> <p>marketing - Indicates the conversation was opened by a business sending template categorized as MARKETING to the customer.</p> <p>utility - Indicates the conversation was opened by a business sending template categorized as UTILITY to the customer. This applies any time it has been more than 24 hours since the last customer message.</p> <p>service - Indicates that the conversation opened by a business replying to a customer within a customer service window.</p> <p>expiration_timestamp - Date when the conversation expires. This field is only present for messages with a `status` set to `sent`.</p>
pricing	Object	<p>category - Indicates the conversation category:</p> <p>authentication - Indicates an authentication conversation.</p> <p>marketing - Indicates a marketing conversation.</p> <p>utility - Indicates a utility conversation.</p> <p>service - Indicates a service conversation.</p>

		pricing_model – Type of pricing model used by the business. Current supported value is CBP
errors	Array of objects	<p>An array of error objects describing the error. Error objects have the following properties, which map to their equivalent properties in API error response payloads.</p> <p>Webhooks triggered by v15.0 and older requests:</p> <ul style="list-style-type: none"> • code – Integer. Example: 130429. • title – String. Error code title. Example: Rate limit hit. <p>Webhooks triggered by v16.0 and newer requests:</p> <ul style="list-style-type: none"> • code – Integer. Error code. Example: 130429. • title – String. Error code title. Example: Rate limit hit. • message – String. Error code message. This value is the same as the title value. For example: Rate limit hit. Note that the message property in API error response payloads pre-pends this value with the a # symbol and the error code in parenthesis. For example: (#130429) Rate limit hit. • error_data – Object. An error data object with the following properties: • details – String. Describes the error. Example: Message failed to send because there were too many messages sent from this phone number in a short period of time.
id	String	The ID for the message that the business that is subscribed to the webhooks sent to a customer
recipient_id	String	The customer's WhatsApp ID. A business can respond to a customer using this ID. This ID may not match the customer's phone number, which is returned by the API as input when sending a message to the customer.
status	String	<p>delivered – A webhook is triggered when a message sent by a business has been delivered</p> <p>read – A webhook is triggered when a message sent by a business has been read</p> <p>sent – A webhook is triggered when a business sends a message to a customer</p>

timestamp	Unix timestamp	Date for the status message
-----------	-------------------	-----------------------------

NOTE: For a status to be **read**, it must have been **delivered**. In some scenarios, such as when a user is in the chat screen and a message arrives, the message is **delivered** and **read** almost simultaneously. In this or other similar scenarios, the delivered notification will not be sent back, as it is implied that a message has been **delivered** if it has been read. The reason for this behavior is internal optimization.

For further assistance or clarification, please contact our support team at support@messagecentral.com.