

WhatsApp

This chapter of our API Documentation is designed for customers who have been granted access to WhatsApp Business API. It is important that you request access to this program via [our Channels portal](#). Without this approval you can not make use of this part of the API.

WhatsApp also comes with a considerable set of 'business rules' that you might have to take into account when using WhatsApp. Be sure to read our [Implementation guide](#) and its [Rules and Regulations](#)

Note: since WhatsApp for Business is used for 2-way communication (chat) it is important to also implement the Inbound flow using our [API documentation of the Inbound webhook](#).

Message encryption

When making use of the WhatsApp Business API solution, messages are encrypted on our platform. In our WhatsApp implementation guide we explain how this works. Encryption is a mandatory WhatsApp feature, implemented by [CM.com](#). You don't have to do anything to enable this, it is part of the the WhatsApp account creation.

Sending a WhatsApp message

For service requests, users usually initiate a conversation and will reach out to you. To link incoming messages to your outbound messages, the phone number field is used.

Message example

This is the most basic form of sending a message via WhatsApp by specifying `allowedChannels` with a `WhatsApp` value. In this example we only make use of the required 'body' attribute, which is backwards compatible with SMS channel

requirements. This however also means it is affected by maximum size limitations of SMS as described in the section on [SMS multipart messaging](#).

```
{
  "messages": {
    "authentication": {
      "productToken": "your product token"
    },
    "msg": [{
      "body": {
        "type": "auto",
        "content": "Message Text"
      },
      "to": [{
        "number": "00316012345678"
      }],
      "from": "TestSender",
      "allowedChannels": ["WhatsApp"]
    }]
  }
}
```

More common is to use the [Rich Content](#) conversation array, with 1 or more conversation items (or speech bubbles) and only use the `body` 's content as fallback content for channels that do not support rich content.

```
{
  "messages": {
    "authentication": {
      "productToken": "your product token"
    },
    "msg": [{
      "body": {
        "type": "auto",
        "content": "Fallback Text"
      },
      "to": [{
        "number": "00316012345678"
      }],
      "from": "TestSender",
      "allowedChannels": ["WhatsApp"],

```

```

    "richText": {
      "conversation": [{
        "text": "A text message with bold formatting in a",
      }, {
        "text": "Another speech bubble"
      }, {
        "media": {
          "mediaName": "and an image",
          "mediaUri": "https://www.cm.com/cdn/web/nl-nl/t",
          "mimeType": "image/jpg"
        }
      }
    ]
  }
}

```

Note: The WhatsApp protocol strives for, but does not guarantee, in-order delivery of your messages.

Text messages

Text messages are and should always be encoded in UTF-8 and do support characters like emojis. This is true for both the `body` field and when using a `text` item inside a [Rich Content](#) conversation.

The maximum length of a WhatsApp text message bubble is 4096 characters.

WhatsApp specifically allows a limited amount of formatting for text messages that you can make use of. These markup characters need to be prefixed with a space character in order for them to take effect.

Formatting	Symbol	Example
Bold	Asterisk (*)	Your total is *\$10.50* .
<i>Italics</i>	Underscore (_)	Welcome to <i>_WhatsApp_!</i>
Strike-through	Tilde (~)	This is better best!
Code	Three backticks (`)	<code>`print 'Hello World';`</code>

A line break can be inserted by using: \n

Message example

```
{
  "messages": {
    "authentication": {
      "productToken": "your product token"
    },
    "msg": [{
      "body": {
        "type": "auto",
        "content": "Fallback Text"
      },
      "to": [{
        "number": "00316012345678"
      }],
      "from": "TestSender",
      "allowedChannels": ["WhatsApp"],
      "richContent": {
        "conversation": [{
          "text": "A text message with *bold* formatting in a",
          "type": "text"
        }, {
          "text": "Another speech bubble",
          "type": "speechBubble"
        }
      ]
    }
  ]
}
```

URL previews

Any URLs present in the message text of a WhatsApp message will be automatically recognized by the clients as links. Additionally, the first URL encountered in a text message will generate a URL preview. For this it is required that the URL starts with

`http://` or `https://`

A URL preview can contain a title, a message and a small fragment of text. These elements are automatically retrieved from the destination website. To support this preview, the destination website needs to have implemented the [Open Graph Protocol](#).

The URL preview is retrieved by the WhatsApp software running on the [CM.com](#) side, before being sent to the client. This is for privacy reasons.

Template

To send a message to a user first a template must be used. These templates must be setup and approved before they can be used. You can request templates via the [Channels portal](#).

Once a template is approved you can use the template namespace and name to send a message (Use the details button in your template overview to find these values).

The template message must have the exact same number of variables as the template has. The same template can be set-up in multiple languages, hence you also need a language parameter.

Please remember that a user must have opted in for the type of communication as well. The opt-in must be an active opt-in. This means it must be triggered by a user action, such as entering a phone number in a WhatsApp field or checking a box to indicate consent.

See also [Facebook's opt-in guide](#).

A template object contains a 'whatsapp' object, which corresponds to the template object in the [WhatsApp for Business docs](#).

Field	Description	Required
namespace	The namespace that will be used, will be provided to you after the template is approved	Yes
element_name	The element name of the template that will be used, will be provided to you after the	Yes

	template is approved	
language	Specifies the language, see below for object definition	Yes
localizable_params	An array of parameters to personalize the templated message. See the WhatsApp documentation for advanced details.	Yes (only for regular templates)
components	Array containing the content-components of the message, to personalize the templates message. See the WhatsApp documentation for advanced details.	Yes (only for media templates)

Language

Field	Description	Required
policy	<p>deterministic — Deliver the message template in exactly the language and locale asked for.</p> <p>fallback — deprecated Deliver the message template in the language that matches user's language/locale setting on device. If one can't be found, deliver using the specified fallback language.</p>	Yes
code	The code of the language or locale to use — Accepts both language and language_locale formats (e.g., en and en_US).	Yes

WhatsApp maintains a [list of valid language codes](#)

Note: Please note that the fallback language policy will be deprecated by WhatsApp in January 2020 and the deterministic language policy will become the default policy.

Example template message

```
{
```

```
"messages": {
  "authentication": {
    "producttoken": "your product token"
  },
  "msg": [{
    "from": "CM.com",
    "to": [{
      "number": "0031612345678"
    }
  ],
  "body": {
    "type": "auto",
    "content": "This is a WhatsApp message"
  },
  "allowedChannels": ["WhatsApp"],
  "richContent": {
    "conversation":
    [{
      "template": {
        "whatsapp": {
          "namespace": "your namespace id here",
          "element_name": "TEMPLATE_NAME",
          "language": {
            "policy": "deterministic",
            "code": "en"
          }
        },
        "localizable_params": [{
          "default": "Name"
        }, {
          "default": "Event"
        }, {
          "default": "Saturday 6, July",
          "date_time": {
            "component": {
              "day_of_week": "Saturday",
              "day_of_month": 6,
              "year": 2019,
              "month": 7,
              "hour": 13,
              "minute": 0
            }
          }
        }
      ]
    }
  ]
}
```

```

    }, {
      "default": "€10.50",
      "currency": {
        "currency_code": "EUR",
        "amount_1000": 10500
      }
    }
  ]
}

```

Example media template message

```
{
  "messages": {
    "authentication": {
      "producttoken": "your product token"
    },
    "msg": [{
      "from": "CM.com",
      "to": [{
        "number": "0031612345678"
      }
    ],
    "body": {
      "type": "auto",
      "content": "This is a WhatsApp message"
    },
    "allowedChannels": ["WhatsApp"],
    "richContent": {
      "conversation":
      [{
        "template": {
          "whatsapp": {
```



```
    "namespace": "your namespace id here",
    "element_name": "TEMPLATE_NAME",
    "language": {
      "policy": "deterministic",
      "code": "en"
    },
    "components": [
      {
        "type": "header",
        "parameters": [
          {
            "type": "image",
            "media": {
              "mediaName": "conversat",
              "mediaUri": "https://ww",
              "mimeType": "image/jpg"
            }
          }
        ]
      },
      {
        "type": "body",
        "parameters": [
          {
            "type": "text",
            "text": "replace-value-1_na"
          },
          {
            "type": "text",
            "text": "replace-value-2_ev"
          }
        ]
      }
    ]
  }
}
```

Media messages

In WhatsApp the following media types are supported:

Type	Supported file types	to send
image	JPG, JPEG, PNG	image/jpeg, image/png
video	MP4 (see notes)	video/mp4
audio	AAC, M4A, AMR, MP3, OGG OPUS	audio/aac, audio/m4a, audio/amr
		audio/mp3, audio/ogg; codecs=opus
document	PDF, DOC(X), PPT(X), XLS(X), etc	Any valid MIME-type

Notes:

- For video only H.264 video codec and AAC audio codec is supported.
- Sending video without audio is not supported
- The media type “voice” is not supported for sending a message
- The media type GIF and stickers are not supported.
- The maximum file size for uploading media is **100MB**.
- After uploading the media is post-processed and there are post-processing limits for the various media types outlined in the [Post-Processing Media Size table](#).

Media object contains the following fields:

Term	Description
MediaName	The filename for a document, or the caption of the image or video (not supported for audio).

MediaUri	The location of the image, audio or video. The Uri should return a correct Content-Type header
MimeType	The mime type of the image, audio or video.

```
{
  "conversation": [
    {
      "media": {
        "mediaName": "conversational-commerce",
        "mediaUri": "https://www.cm.com/cdn/web/nl-nl/blog/conv",
        "mimeType": "image/jpg"
      }
    }
  ]
}
```

As data usage costs might apply when the end user is not on Wifi, please be considerate when sending media messages. We strongly advise not to send files larger than 1 MB, and for a conversation not to exceed 20Mb.

Location messages

To send a location to a user use the location object, which contains the following fields:

Latitude	The latitude in degrees
Longitude	The longitude in degrees
SearchQuery	The address to display
Label	The label to display at the pin

Example

```
{
```

```

    "conversation": [
      {
        "location": {
          "latitude": "51.603802",
          "longitude": "4.770821",
          "label": "CM HQ",
          "searchQuery": "Konijnenberg 30"
        }
      }
    ]
  }
}

```

Contacts

To send one or more contacts to a user use the Contacts array, which contains the fields described in [WhatsApp for Business docs](#).

Note that the `name.formatted_name` and at least one of: `name.first_name`, `name.last_name`, `name.middle_name`, `name.suffix`, `name.prefix` is required.

Example

```

{
  "conversation": [
    {
      "contacts": [
        {
          "addresses": [
            {
              "city": "Breda",
              "country": "Netherlands",
              "country_code": "NL",
              "street": "Konijnenberg 30",
              "type": "WORK",
              "zip": "4825 BD"
            }
          ],
          "name": {
            "formatted_name": "CM Developer",

```

```
"last_name": "Your last name"
```

```
}
```

```
}
```

```
]
```

```
}
```

```
]
```

```
}
```