



SandeshLive



Services

Sending Media Messages

Use the messages node to send messages containing audio, documents, images, stickers, or videos to your customers.

In essence, when you send a message that includes media, you must provide either the ID of the uploaded media or a link to the media in the request body. You must also specify the type of media that you are sending: audio, document, image, sticker, or video. When the request is received, the media is uploaded to the WhatsApp server and sent to the user indicated in the to field.

Currently, there are two ways to send media messages with the WhatsApp Business API:

- **IDs** — To use an ID, you must first upload the media using the [media node](#) to obtain the ID required for the messages API call.
- **Links** — To use a link, you supply an HTTP(S) link from which the application will download the media, saving you the step of uploading media yourself.

Before You Start

You need:

- To meet all the prerequisites listed in the [Before You Start section](#) of the [Messages documentation](#).
- To upload the media you're sending or have a link to it.

Step 1: Make a POST Request to /messages

After you [upload the media](#), use the returned ID for the id field in the API call sending the media message. Alternatively, you can provide a link parameter pointing to the media you want to send (currently only HTTP/HTTPS links are supported).

Either id or link is required, but should not be used at the same time.

Example



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

POST /v1/messages

```
{
  "recipient_type": "individual",
  "to": "whatsapp-id",
  "type": "audio" | "contact" | "document" | "image" | "location" | "sticker" | "text" | "video",

  "audio": {
    "id": "your-media-id",
  }

  "document": {
    "id": "your-media-id",
    "caption": "your-document-caption-to-be-sent",
    "filename": "your-document-filename"
  }

  "document": {
    "link": "the-provider-name/protocol://the-url",
    "provider": {
      "name": "provider-name"
    },
    "caption": "your-document-caption"
  }

  "document": {
    "link": "http(s)://the-url.pdf",
    "caption": "your-document-caption"
  }

  "video": {
    "id": "your-media-id",
    "caption": "your-video-caption"
  }

  "image": {
    "link": "http(s)://the-url",
    "provider": {
      "name": "provider-name"
    },
    "caption": "your-image-caption"
  }
}
```



```
}  
  
"image": {  
  "id": "your-media-id",  
  "caption": "your-image-caption"  
}  
  
"sticker": {  
  "id": "your-media-id"  
}  
  
"sticker": {  
  "link": "http(s)://the-url",  
  "provider": {  
    "name": "provider-name"  
  }  
}  
}
```

For more information on parameters, see:

- [Parameters common to all message requests](#)
- [Parameters for media messages](#)
- [The media object](#)

Step 2: Check Your Response

The successful response includes a messages object with a message ID.

```
{  
  "messages": [{  
    "id": "gBEGkYiEB1VXAgIK1ZEqA1YKPrU"  
  }]  
}
```

In the case of an unsuccessful response, a callback is sent to your Webhook URL even though the response will yield a message ID similar to a successful message send. This is why it's important to have a [Webhook](#) server set up.

See [Error and Status Codes](#) for more information on errors.