

RH-Whatsapp-API v1.0.0

GETTING STARTED

Introduction

Welcome to Reply Hero API documentation. You can interact with the API through HTTP requests from any language.

Authentication

The ReplyHero API uses API keys for authentication. Visit our API Keys endpoint to create the API key you'll use in your requests.

Remember that your API key is a secret! Do not share it with others or expose it in any client-side code (browsers, apps). Production requests must be routed through your own backend server where your API key can be securely loaded from an environment variable or key management service.

```
X-API-KEY: REPLYHERO_API_KEY
```

Example curl command:

```
curl --location 'https://whatsapp.replyherotech.com/whatsapp/messages/send' \
-H 'X-API-KEY: eUG3uKs0r6YoHmkRZVXe0nCemJY19r'
```

Making Requests

You can paste the command below into your terminal to run your first API request. Make sure to replace \$REPLYHERO_API_KEY with your secret API key.

```
curl --location 'https://whatsapp.replyherotech.com/whatsapp/messages/send' \
--header 'x-api-key: eUG3uKs0r6YoHmkRZVXe0nCemJY19r' \
```

```

--header 'Content-Type: application/json' \
--data '{
  "message": {
    "type": "text",
    "text": {
      "body": "testing"
    }
  },
  "sessionId": 'YOUR_SESSEION_ID',
  "customerId": 'YOUR_CUSTOMER_ID',
  "userId": 'YOUR_USER_ID',
  "orgId": 'YOUR_ORG_ID'
}'

```

Customer IDs is the receiver IDs ***(needs to request from RH)**

User IDs is the admin of the organization ***(needs to request from RH)**

Organization IDs ***(needs to request from RH)**

This request queries the `send WhatsApp message` endpoint to send a message with a text of **"testing"**. You should get a response back the resembles the following:

```

{
  "customerId": "CUSTOMER_ID",
  "id": "MESSAGE_ID",
  "messageType": "human",
  "orgId": "ORG_ID",
  "response": {
    "contacts": [
      {
        "input": "601*****67",
        "wa_id": "601*****67"
      }
    ],
    "messages": [
      {
        "id":
"wamid.HBgLNjAxODM4OTE2NjcVAgARGBI3QzVBREE1Qzg5NTRCREY1RjIA"
      }
    ],
    "messaging_product": "whatsapp"
  }
}

```

```
    },  
    "sessionId": "testing",  
    "timestamp": "Sun, 17 Mar 2024 00:45:25 +0800",  
    "userId": "UKIR5Iumv6a6fs1ULT6BKBmyMjG3"  
  }  
}
```

***note** that the client will only receive the message if they initiated the conversation, and the conversation window remain for **24 hours** only

Now that you've send your first message in WhatsApp, let's break down the **response object**. We can see that `messageType` is `human` which means the WhatsApp message is sent through human. There are few `messageType` such as `human`, `user`, `assistant`, and `template`.

`human`, `template`, and `assistant` here are all classified as **sender**; while `user` here is the receiver.

Below show case some use cases:

1. **Customer Service A (human)** reply WhatsApp message to **Customer B**
 - `human`: **Customer Service A** (the sender)
 - `user`: **Customer B** (the receiver)
2. **Customer Service A (bot)** reply WhatsApp message to **Customer B**
 - `assitant`: **Customer Service A** (the sender)
 - `user`: **Customer B** (the receiver)
3. **Customer Service A (template)** send WhatsApp template message to **Customer B**
 - `template`: **Customer Service A** (the sender)
 - `user`: **Customer B** (the receiver)

ENDPOINTS

Send WhatsApp Message

```
POST https://whatsapp.replyherotech.com/whatsapp/messages/send
```

Sends a message to a specified recipient via WhatsApp.

Text Message

Programmatically send a text messages via Whatsapp.

Request Body

message object **Required**

An object of messages comprising the type and body of the message.

Text Message >

Text message object

properties

type string **Required**

The type of the message, in this case `text`

text object **Required**

The type of the message, in this case `text`

- **body** string **Required**

The body message to send

sessionId string **Required**

The session IDs for this communication session

customerId string **Required**

The customer IDs for the receiver

***Required from RH**

userId string **Required**

The admin of the organization

***Required from RH**

orgId string **Required**

The organization IDs for the sender

***Required from RH**

Example Requests

Example curl commnd:

```
curl --location 'https://whatsapp.replyherotech.com/whatsapp/messages/send' \
--header 'x-api-key: REPLYHERO_API_KEY' \
--header 'Content-Type: application/json' \
--data '{
  "message": {
    "type": "text",
    "text": {
      "body": "testing msg"
    }
  },
  "sessionId": "SESSION_ID",
  "customerId": "CUSTOMER_ID",
  "userId": "USER_ID",
  "orgId": "ORG_ID"
}'
```

Example Python - request command

```
import requests
import json

url = "https://whatsapp.replyherotech.com/whatsapp/messages/send"
```

```

payload = json.dumps({
    "message": {
        "type": "text",
        "text": {
            "body": "testing msg"
        }
    },
    "sessionId": "SESSION_ID",
    "customerId": "CUSTOMER_ID",
    "userId": "USER_ID",
    "orgId": "ORG_ID"
})
headers = {
    'x-api-key': 'REPLYHERO_API_KEY',
    'Content-Type': 'application/json'
}

response = requests.request("POST", url, headers=headers, data=payload)

print(response.text)

```

Response Example

Example success response

```

{
    "customerId": "CUSTOMER_ID",
    "id": "MESSAGE_ID",
    "messageType": "human",
    "orgId": "ORG_ID",
    "response": {
        "contacts": [
            {
                "input": "601*****67",
                "wa_id": "601*****67"
            }
        ],
        "messages": [
            {
                "id": "wamid.HBgL*****REY1RjIA"
            }
        ]
    }
}

```

```
    }  
  ],  
  "messaging_product": "whatsapp"  
},  
"sessionId": "SESSION_ID",  
"timestamp": "Sun, 17 Mar 2024 00:45:25 +0800",  
"userId": "USER_ID"  
}
```

Media Message

Programmatically send a media messages via WhatsApp.

Request Body

message object **Required**

An object of messages comprising the type and body of the message.

Media Message >

Media message object

properties

type string **Required**

The type of the message, in this case `image`.

Supported media type include:

`image`, `document`, and `video`.

Media object **Required**

The type of the message, in this case `image`

Supported media type include:

`image`, `document`, and `video`.

☰ Media Object >

Media object

properties

file object **Optional**

The image file to be sent.

Required when `type` is `document`, `image`, or `video` and you are not using `link`.

link string **Optional**

The protocol and URL of the media to be sent. Use **only** with HTTP/HTTPS URLs.

Required when `type` is `document`, `image`, or `video` and you are not using `file`.

caption string **Optional**

The media asset caption

filename string **Optional**

Describes the filename for the specific document. Use **only** with `document` media.

provider string **Optional**

This path is optionally used with a `link` when the HTTP/HTTPS link is not directly accessible and requires additional configurations like a bearer token.

☰ Media File >

File object

properties

name string **Required**

The name of the image file to be sent

type string **Required**

Type of media file being uploaded. See [Supported Media Types](#) for more information.

displayType string **Required**

The type of the media to be displayed

Supported displayType **include:**

IMAGE , DOCUMENT , and VIDEO .

extension string **Required**

The extension of the image file to be sent

data string **Required**

The **bytes** of the image file to be sent in string

sessionId string **Required**

The session IDs for this communication session

customerId string **Required**

The customer IDs for the receiver

*Required from RH

userId string **Required**

The admin of the organization

*Required from RH

orgId string **Required**

The organization IDs for the sender

*Required from RH

Example Requests

Example curl command:

```
curl --location 'https://whatsapp.replyherotech.com/whatsapp/messages/send' \
--header 'x-api-key: REPLYHERO_API_KEY' \
--header 'Content-Type: application/json' \
--data '{
  "message": {
    "type": "image",
    "image": {
      "file": {
        "name": "bot.png",
        "type": "image/png",
        "displayType": "IMAGE",
        "extension": "png",
        "data": "BYTES_DATA_IN_STRING"
      },
      "caption": "testing"
    }
  },
  "sessionId": "SESSION_ID",
  "customerId": "CUSTOMER_ID",
}
```

```
"userId": "USER_ID",  
"orgId": "ORG_ID"  
}
```

Example Python - request command:

```
import requests  
import json  
  
url = "https://whatsapp.replyherotech.com/whatsapp/messages/send"  
  
payload = json.dumps({  
    "message": {  
        "type": "image",  
        "image": {  
            "file": {  
                "name": "bot.png",  
                "type": "image/png",  
                "displayType": "IMAGE",  
                "extension": "png",  
                "data": "BYTES_DATA_IN_STRING"  
            },  
            "caption": "testing"  
        }  
    },  
    "sessionId": "SESSION_ID",  
    "customerId": "CUSTOMER_ID",  
    "userId": "USER_ID",  
    "orgId": "ORG_ID"  
})  
headers = {  
    'x-api-key': 'REPLYHERO_API_KEY',  
    'Content-Type': 'application/json'  
}  
  
response = requests.request("POST", url, headers=headers, data=payload)  
  
print(response.text)
```

Response Example

Example success response

```
{
  "customerId": "CUSTOMER_ID",
  "id": "MESSAGE_ID",
  "messageType": "human",
  "orgId": "ORG_ID",
  "response": {
    "contacts": [
      {
        "input": "601*****67",
        "wa_id": "601*****67"
      }
    ],
    "messages": [
      {
        "id": "wamid.HBgL*****REY1RjIA"
      }
    ],
    "messaging_product": "whatsapp"
  },
  "sessionId": "SESSION_ID",
  "timestamp": "Sun, 17 Mar 2024 00:45:25 +0800",
  "userId": "USER_ID"
}
```

Send WhatsApp Template

POST <https://whatsapp.replyherotech.com/whatsapp/template/send>

Sends a template message to all the recipients that are grouped under the same organization via WhatsApp.

Template Message

Programmatically send a template message via WhatsApp.

Request Body

templateName string **Required**

An object of messages comprising the type and body of the message.

sessionId string **Required**

The session IDs for this communication session

userId string **Required**

The admin of the organization

Required from RH

orgId string **Required**

The organization IDs for the sender

***Required from RH**

components string **Required**

A list of components objects comprising the parameters of the message.

 **Components Object** >

Components Object object

properties

type string **Required**

The type of the component.

Supported component type include:

header , and body .

Parameters array **Required**

A list of parameters objects with the content of the message

Parameters Object >

Parameters object

properties

type string **Required**

The type of the parameters.

Supported `type` is `text`, `document`, `image`, and `video`.

***Note:** only component `type` `header` support `document`, `image`, and `video`.

text string **Optional**

The message's text. Character limit varies based on the following included component type.

For the `header` component `type`:

- 60 characters

For the `body` component `type`:

- 1024 characters if other component types are included
- 36768 characters if `body` is the only component type included

Required when `type` is `text`.

Media Object object **Optional**

The type of the media, in this case `image`

Supported media type include:

image, document, and video.

Required when type is image, document, or video.

☰ Media Object >

Media object

properties

id string Optional

The media object ID. Do not use this field when parameter type is set to text.

Required when parameter type is document, image, or video and you are not using link.

link string Optional

The protocol and URL of the media to be sent. Use **only** with HTTP/HTTPS URLs.

Required when parameter type is document, image, or video and you are not file.

caption string Optional

The media asset caption

filename string Optional

Describes the filename for the specific document. Use **only** with document media.

provider string Optional

This path is optionally used with a link when the HTTP/HTTPS link is not directly accessible and requires additional configurations like a bearer token.

Example Requests

Example curl command:

```
curl --location 'https://whatsapp.replyherotech.com/whatsapp/template/send' \
--header 'x-api-key: REPLYHERO_API_KEY' \
--header 'Content-Type: application/json' \
--data '{
  "templateName": "TEMPLATE_NAME",
  "sessionId": "SESSION_ID",
  "orgId": "ORG_ID",
  "userId": "USER_ID",
  "components": [
    {
      "type": "header",
      "parameters": [
        {
          "type": "image",
          "image": {"id": "MEDIA_ID"}
        }
      ]
    },
    {
      "type": "body",
      "parameters": [
        {
          "type": "text",
          "text": "https://replyhero.io/"
        },
        {
          "type": "text",
          "text": "https://replyhero.io/"
        }
      ]
    }
  ]
}'
```

Example Python - request command:


```
import json

url = "https://whatsapp.replyherotech.com/whatsapp/template/send"

payload = json.dumps({
    "templateName": "TEMPLATE_NAME",
    "sessionId": "SESSION_ID",
    "orgId": "ORG_ID",
    "userId": "USER_ID",
    "components": [
        {
            "type": "header",
            "parameters": [
                {
                    "type": "image",
                    "image": {
                        "id": "MEDIA_ID"
                    }
                }
            ]
        },
        {
            "type": "body",
            "parameters": [
                {
                    "type": "text",
                    "text": "https://replyhero.io/"
                },
                {
                    "type": "text",
                    "text": "https://replyhero.io/"
                }
            ]
        }
    ]
})

headers = {
    'x-api-key': 'REPLYHERO_API_KEY',
    'Content-Type': 'application/json'
}

response = requests.request("POST", url, headers=headers, data=payload)

print(response.text)
```

Response Example

Example success response

```
[
  {
    "customerId": "CUSTOMER_ID",
    "file": null,
    "id": "MESSAGE_ID",
    "message": "TEMPLATE_NAME",
    "messageType": "template",
    "orgId": "ORG_ID",
    "response": {
      "contacts": [
        {
          "input": "601*****13",
          "wa_id": "601*****13"
        }
      ],
      "messages": [
        {
          "id": "wamid.HBgL*****REY1RjIA"
        }
      ],
      "messaging_product": "whatsapp"
    },
    "sessionId": "SESSION_ID",
    "timestamp": "Sun, 17 Mar 2024 13:37:13 +0800",
    "userId": "USER_ID"
  },
  {
    "customerId": "CUSTOMER_ID",
    "file": null,
    "id": "MEESSAGE_ID",
    "message": "TEMPLATE_NAME",
    "messageType": "template",
    "orgId": "ORG_ID",
    "response": {
```

```
    "contacts": [
      {
        "input": "601*****67",
        "wa_id": "601*****67"
      }
    ],
    "messages": [
      {
        "id": "wamid.HBgL*****REY1RjIA"
      }
    ],
    "messaging_product": "whatsapp"
  },
  "sessionId": "SESSION_ID",
  "timestamp": "Sun, 17 Mar 2024 13:37:13 +0800",
  "userId": "USER_ID"
},
{
  "customerId": "CUSTOMER_ID",
  "file": null,
  "id": "MESSAGE_ID",
  "message": "TEMPALTE_NAME",
  "messageType": "template",
  "orgId": "ORG_ID",
  "response": {
    "contacts": [
      {
        "input": "601*****28",
        "wa_id": "601*****28"
      }
    ],
    "messages": [
      {
        "id": "wamid.HBgL*****REY1RjIA"
      }
    ],
    "messaging_product": "whatsapp"
  },
  "sessionId": "SESSION_ID",
  "timestamp": "Sun, 17 Mar 2024 13:37:14 +0800",
```

```
    "userId": "USER_ID"
  },
  {
    "customerId": "CUSTOMER_ID",
    "file": null,
    "id": "USER_ID",
    "message": "TEMPLATE_NAME",
    "messageType": "template",
    "orgId": "ORG_ID",
    "response": {
      "contacts": [
        {
          "input": "601*****78",
          "wa_id": "601*****78"
        }
      ],
      "messages": [
        {
          "id": "wamid.HBgL*****REY1RjIA"
        }
      ],
      "messaging_product": "whatsapp"
    },
    "sessionId": "SESSION_ID",
    "timestamp": "Sun, 17 Mar 2024 13:37:15 +0800",
    "userId": "USER_ID"
  }
]
```

Get Media ID

POST https://whatsapp.replyherotech.com/whatsapp/uploads/media_upload

To complete some of the API calls such as [Send WhatsApp Template](#), you need to have a media ID. You can retrieve the media ID but uploading the media.

Upload Media Once you have successfully uploaded media files to the API, the media ID is included in the response to your call.

Request Body *(form-data)

orgId string **Required**

The organization IDs for the sender

*Required from RH

type string **Required**

Type of media file being uploaded. See [Supported Media Types](#) for more information.

file file **Required**

The media file

Example Requests

Example curl command:

```
curl --location
'https://whatsapp.replyherotech.com/whatsapp/uploads/media_upload' \
--header 'X-API-KEY: REPLYHERO_API_KEY' \
--form 'orgId="ORG_ID"' \
--form 'type="image/jpg"' \
--form 'file=@"/path/to/image/example.jpeg"'
```

Example Python - request command:

```
import requests

url = "https://whatsapp.replyherotech.com/whatsapp/uploads/media_upload"

payload = {'orgId': 'ORG_ID',
```

```
'type': 'image/jpg'}
files=[
    ('file',
 ('example.jpeg', open('/path/to/image/example.jpeg', 'rb'), 'image/jpeg'))
]
headers = {
    'X-API-KEY': 'REPLYHERO_API_KEY'
}

response = requests.request("POST", url, headers=headers, data=payload,
files=files)

print(response.text)
```

Response Example

Example success response

```
{
  "id": "1063108041439098"
}
```

Supported Media Types

Supported options for images are:

- image/jpeg
- image/png

Supported options for documents are:

- text/plain
- application/pdf
- application/vnd.ms-powerpoint
- application/msword
- application/vnd.ms-excel
- application/vnd.openxmlformats-officedocument.wordprocessingml.document

- `application/vnd.openxmlformats-officedocument.presentationml.presentation`
- `application/vnd.openxmlformats-officedocument.spreadsheetml.sheet`

Supported options for video are:

- `video/mp4`
- `video/3gp`