MIGAWHATSAPP SEND MESSAGE & Chat History API Documentation

1. Introduction

The API class is a part of the Migawhatsapp module and is responsible for handling the sending of messages through the Migawhatsapp Official Send Message API. This class performs various checks and validations, such as verifying security tokens, validating phone numbers and email addresses, and ensuring the existence and validity of files. It supports different media types, including images, files, audio, and video, and logs the text-to-speech (TTS) process details. The class ensures messages are sent securely and efficiently, adhering to the specified constraints and configurations.

We have an endpoint `checkphone` that will help the user to verify whether the phone number exists on WhatsApp or not. Also, this feature is already implemented in the `message` API.

We have just updated the `message` endpoint to support ***group messages***. All you need to do is add 2 extra parameters to the API, and you are done. The `is_sent_to_group` parameter will be used to tell the API that we want to send this message to a group, and the value can be `1` to make this happend. When `is_sent_to_group = 1`, you will be required to send another required parameter called `group_id`, which will tell the API about the group you want to send a message to. The `group_id` will become a required parameter, and note that now `phone` parameter won't be required anymore because it does not make sense in case of group messages.

Now the question is, how will you get this group ID? No worry, we got you covered with our new endpoint called `/migawhatsapp/api_send/groups`.

1.1. API Endpoints

As access points to our API service, we have three endpoints:

1.1.1. Heartbeat Endpoint (`heartbeat`)

- `POST` Request URL:
  - `https://yourdomain.com/migawhatsapp/api_send/heartbeat`

1.1.2. Message Endpoint (`message`)

- `POST` Request URL:
    - `https://yourdomain.com/migawhatsapp/api_send/message`

1.1.3. Check Phone Endpoint (`checkphone`)

- `POST` Request URL:
    - `https://yourdomain.com/migawhatsapp/api_send/checkphone`

1.1.4. Get All Groups Endpoint (`groups`)

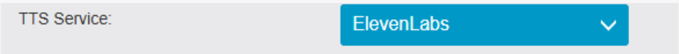
- `POST` Request URL:
    - `https://yourdomain.com/migawhatsapp/api_send/groups`
    - !!IMPORTANT!!: Group listing is only available when using the `WA API MASTER`. `WA API SLAVE` does not support group queries.

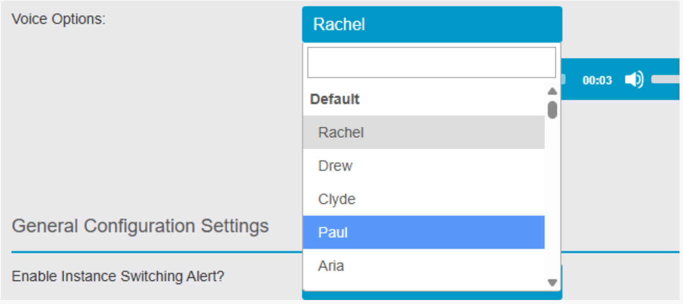
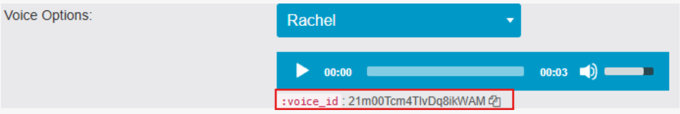
1.2. API Parameters

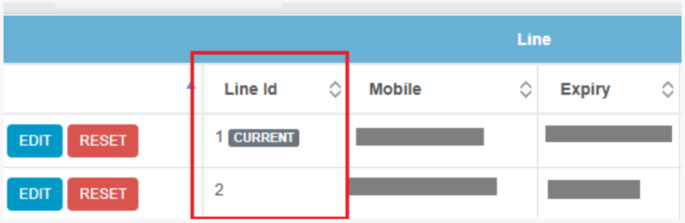
Both `heartbeat` and `message` endpoints use the same parameters, but the `checkphone` endpoint only requires 3 parameters [`security_token`, `app_id`, `phone`], and the new `groups` endpoint only requires 3 parameters [`security_token`, `app_id`, `line_id (optional)`] from the list below:

Parameter	Type	Description
app_id	Int	**Mandatory** - YOUR_APP_ID (seen from the module's dashboard at the top left).
security_token	String	**Mandatory** - YOUR_SECURITY_TOKEN (this token is generated from the API Settings section under the module's Settings tab with the label Send Message API Token? Make sure you have enabled the send message API as well.)
message	String	**Mandatory (conditional)** - The message you want to send. This parameter is optional when we have a `file_url` parameter present because a file can be sent without any `message`.
phone	String	● **Rules**

		○ Send a phone number of 10-14 digits with + international country code at the beginning ○ If the phone number exists in the database already, you will also receive `sib_user_id` in the response ● **Optional** ○ The phone number can be optional if you have sent the email parameter of the user that exists in the platform as a registered user and has a valid phone number added to his/her profile ● **Mandatory** ○ The phone number will be required if we don't send the email parameter In this case, the message is always sent, as we always need a valid phone number to send a message. In the case of `groups` endpoint call, this parameter is `not` required at all.
email	String	● **Rules** ○ Send a valid email address that must exist in the database (and there should be a phone number attached to that account in the database) ○ If the email exists in the database already, you will also receive `sib_user_id` in the response ● **Optional** ○ Email can be optional if you have sent the `phone` number of the user who exists in the platform as a registered user and has a valid email address added to his/her profile ● **Mandatory** ○ An email will be required if we don't send the phone parameter ○ In this case, the message will only be sent if the user exists and has a valid phone number attached In the case of `groups` endpoint call, this parameter is `not` required at all.
name	String	**Optional** - First name of the user (if you want to register the user as well)

surname	String	**Optional** - Last name of the user (if you want to register the user as well)
file_url	URL	**Optional** - Stores the information about the uploaded file URL (you can use any publicly accessible file URL that is supported on WhatsApp) **Note:** Sending audio files is only possible when the PRIMARY/MASTER gateway is working.
tts_enabled	Int	**Optional** - Text to speech (TTS): Now the API supports the text-to-speech feature, which enables you to convert your text message to a selected voice (audio) using the `tts_voice` parameter. Possible values are: ● 1 - enabled ● 0 - disabled (default) If you set the `tts_enabled` value to 1, then the message will be converted to voice or audio and sent to the respective user as an audio message instead of text **Note:** The feature works only when the PRIMARY/MASTER gateway is working. NEW IN V5.2.0  TTS Service: ElevenLabs Now you can select b/w OpenAI (default) and the new one called **ElevenLabs** for TTS Service. You can update your settings from *Settings (Tab) -> OpenAI Configuration -> TTS Service (dropdown).* Migawhatsapp API now supports ElevenLabs for TTS only. You can get the real voices from your creations and use them!
tts_voice	Sting	**Optional** - Different voice options are available for you to select, and these options can be seen [here](): Possible values are (for OpenAI Only): ● alloy - default ● echo ● fable ● onyx ● nova ● shimmer

		 Voice Options: Rachel 00:03 Default Rachel Drew Clyde Paul Aria General Configuration Settings Enable Instance Switching Alert? **ElevenLabs** support different voice types and the type of voice to filter by. One of **'personal'**, 'community', 'default', 'workspace', 'non-default'. 'non-default' is equal to all but 'default'. Personal holds your custom voice that you have created from the ElevenLabs dashboard. You can get the voice parameter value from the **voice_id** text just below the player of the selected voice:  Voice Options: Rachel :voice_id : 21m00Tcm4TlvDq8ikWAM Just click the copy button and use that value; this value will be updated based on the selected value of the Voice Options drop-down.
tts_speed	Float	**Optional** - Different speed ranges are available for voice to select, and these details can be seen here. The possible range is from 0.25 to 4.0. **1.0 is the default (only for OpenAI)**. The tts_speed value must be between **0.1** and **1.2** for **ElevenLabs**. Adjusts the speed of the voice. A value of 1.0 is the default speed, while values less than 1.0 slow down the speech, and values greater than 1.0 speed it up, up to a maximum of 1.2.
line_id	Int	**Optional** - You can set a line_id parameter to specify the line that you want to use to send a message. The line_id has to be valid, and **line_id = 1 is the default if the parameter is not posted at all**. You can get line_id from **Gateway Settings (tab) -> Gateway Settings (section) -> Whatsapp Line Management (grid) -> Under Line ID Column:**

		 Line Line Id Mobile Expiry EDIT RESET 1 CURRENT [redacted] [redacted] EDIT RESET 2 [redacted] [redacted] In the case of `groups` endpoint call, this parameter is **optional**. If you do not post that, the line ID `1` will be used.
is_link_preview_enabled	Int	**Optional** - You can now check if you want to show the preview of the URL link to the receiver or not. Possible values are: ● 1 [ENABLED - **DEFAULT**] ● 0 [DISABLED] **Note**: This is only available in WA API MASTER.
is_sent_to_group	Int	The `is_sent_to_group` parameter will be used to tell the API that we want to send this message to a group, and the value can be `1` to make this happend. When `is_sent_to_group = 1`, you will be required to send another required parameter called `group_id`, which will tell the API about the group you want to send a message to. Possible values are: ● `1` = ENABLED ● `0` OR Not Posted = DISABLED
group_id	String	The `group_id` will become a required parameter, and note that now ~~phone~~ parameter won't be required anymore because it does not make sense in case of group messages. Possible value format could be something like this: ● 123456789325565656@g.us

1.2.1. List of API calls unsuccessful responses and details:

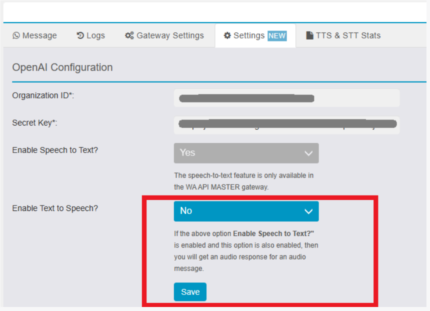
Response	Description

`{"data":null,"error":"No parameter(s) were sent."}`	No required parameter is posted
`{"data":null,"error":"Security token missing."}`	Security token missing.
`{"data":null,"error":"App id missing."}`	App id missing.
`{"data":null,"error":"Invalid app id."}`	Invalid app id.
`{"data":null,"error ":"API is disabled."}`	API is disabled (from the settings).
`{"data":null,"error ":"Security token missing."}`	Security token missing.
`{"data":null,"error ":"Security token mismatched."}`	The security token was mismatched (the wrong token was used).
`{"data":null,"error ":"Gateway settings do not exist."}`	Gateway settings do not exist.
`{"data":null,"error ":"Your gateway settings are no longer working. Please check and reconnect your instance."}`	Your gateway settings are no longer working. Please check and reconnect your instance.
`{"data":null,"error ":"Message is missing."}`	The message is missing.
`{"data":null,"error ":"You must provide either email or phone number."}`	Neither email nor phone is sent or empty.
`{"data":null,"error ":"The phone number is not correct. Please add a phone between 10-14 digits with + international country code at the beginning."}`	The phone number is invalid.
`{"data":null,"error ":"The email address is invalid."}`	The email address is invalid.
`{"data":null,"error ":"No such account exists with this email and phone pair."}`	If you are seeing this error, then there is one scenario: ● You have sent both email and phone parameters, and we did not find this pair inside the database because when you send both email and phone, the API try to check the user's existence first, not the sending of the message

	However, if you send `name` & `surname` with all the details and your account does not exist already, then a new user will be created, and you will receive `sib_user_id` in the response
`{"data":null,"error ":"The phone number associated with this account is not correct. Please add a phone between 10-14 digits with + international country code at the beginning. The current one is: API_SENT_PHONE_NUMBER"}`	This error will appear when you send only the email parameter without the `phone` parameter, and the account is found in the system, but the associated phone number is invalid.
`{"data":null,"error ":"No phone number is associated with this account."}`	This error will appear when you send only the email parameter without the phone parameter, and the account is found in the system, but there is no phone number attached to that account.
`{"data":null,"error ":"No such account exists with this email."}`	This error will appear when you send only the email parameter without the phone parameter, and the account is not found with that email.
`{"data":null,"error ":"We are sorry but this account is already linked to one of our customers."}`	This error will appear when you send only the email parameter without the phone parameter, but with `name` and `surname` parameters and when the system tries to save it finds that the email is already associated with another customer's account.
`{"data":null,"error ":"We are sorry but this account is already linked to one of our customers."}`	This error will appear when you send only the email parameter without the phone parameter, but with `name` and `surname` parameters and when the system tries to save it finds that the email is already associated with another customer's account.
`{"data":null,"error ":"The file does not exist."}`	When the parameter `file_url` has a value, but the file does not exist for the respective URL.

{"data":null,"error ":"The file size is too large. The maximum file size allowed is 25 MB."}	When the parameter file_url has a value, but the file size is > 25MB this error appears. Because the max size allowed is 25MB.
{"data":null,"error ":"The file type is not supported."}	When the parameter file_url has a value that does not lie in the following 4 categories: 1. Image a. image/jpeg b. image/jpg c. image/png d. image/gif i. Only in WA API SLAVE 2. Video a. video/mp4 b. video/ogg c. application/octet-stream 3. Audio a. audio/mpeg b. audio/ogg 4. File a. application/pdf b. application/msword c. application/vnd.openxmlformats-officedocument.wordprocessingml.document d. application/vnd.ms-excel e. application/vnd.openxmlformats-officedocument.spreadsheetml.sheetapplication/vnd.ms-powerpoint f. application/vnd.openxmlformats-officedocument.presentationml.presentation g. text/plain h. text/html i. text/css j. application/ics
{"data":null,"error ":"Sorry PRIMARY API is down and we cannot send native AUDIO files."}	When the parameter file_url has a value, but the PRIMARY/MASTER gateway is down or not set up at all.

	Note: Sending audio files is only possible when the PRIMARY/MASTER gateway is working.
`{"data":null,"error ":"The tts_voice value is not valid."}`	When the parameter `tts_voice` has a value that is not valid or inside the given option, you will see this error.
`{"data":null,"error ":"The tts_speed value must be between 0.25 and 4.0."}`	When the parameter `tts_speed` has a value that is not valid or inside the given range, you will see this error.
`{"data":null,"error ":"Line ID is invalid."}`	When the parameter `line_id` has an invalid value, you will see this error.
`{"data":null,"error ":"You cannot send a message because your line status is EXPIRED|DISABLED|NOT_CONFIGURED."}`	When the selected `line_status` parameter is not in `CONFIGURED` status, you will see this message.
`{"data":null,"error ":"You have reached the maximum limit of messages for this month."}`	When the maximum limit per month for the messages is reached, you will see this message. This includes all types of successful outgoing messages. The default maximum limit is set to `10,000` messages per month.
`{"data":null,"error ":"The is_link_preview_enabled value is not valid."}`	When the value for this parameter is other than 0 or 1.
`{"data":null,"error ":"Gifs are currently only supported by SLAVE."}`	Gifs are currently only supported by SLAVE. You cannot send GIF images using the WA API MASTER as it's only supported in the SLAVE API
`{"data":null,"error ":"Text to Speech is not enabled for this application."}`	If you set the `tts_enabled` value to 1, and from the MAB Migawhatsapp, the settings for TTS are disabled or not yet added, then you will get this error.

	 Message Logs Gateway Settings Settings NEW TTS & STT Stats Save OpenAI Configuration Organization ID*: [redacted] Secret Key*: [redacted] Enable Speech to Text? Yes The speech-to-text feature is only available in the WA API MASTER gateway. Enable Text to Speech? No If the above option **Enable Speech to Text?"** is enabled and this option is also enabled, then you will get an audio response for an audio message.
`{"data":null,"error ":"The phone number does not exist on WhatsApp."}`	Now, when you try to send a message to a phone number that does not exist on WhatsApp, you will get this error message.
`{"data":null,"error ":"The tts_speed value must be between 0.1 and 1.0 for ElevenLabs."}`	The tts_speed value must be between 0.1 and 1.0 for **ElevenLabs**.
`{"data":null,"error ":"Group id missing."}`	When you `is_sent_to_group = 1` and the `group_id` parameter is missing or no value is posted.
`{"data":null,"error ":"Group not found."}`	When you `is_sent_to_group = 1` and the `group_id` parameter's value is not a valid group ID. This error will only appear in case of `WA API MASTER`, as ~~`WA API SLAVE`~~ does not support the groups endpoint and this validation as well.
`{"data":null,"error ":"Group listing is only available when using WA API MASTER. WA API SLAVE does not support group queries."}`	!!IMPORTANT!!: Group listing is only available when using the `WA API MASTER`. ~~`WA API SLAVE`~~ does not support group queries.
`{"data":null,"error ":"Service is temporarily unavailable, please try later."}`	This message will be displayed when the `tts_enabled` parameter is set to `1` and the real-time verification of the connection check for the `WA API MASTER` has failed.
[NEW IN 6.4.0] `{"data":null,"error ":"The service is currently unavailable due to an issue with the WA API Master. Please try again later."}`	**[NEW IN 6.4.0]** If, for some reason, the WA API MASTER server is down, then you might get this response in the `checkphone` endpoint.

1.2.2. List of API calls successful responses and details:

Response	Description

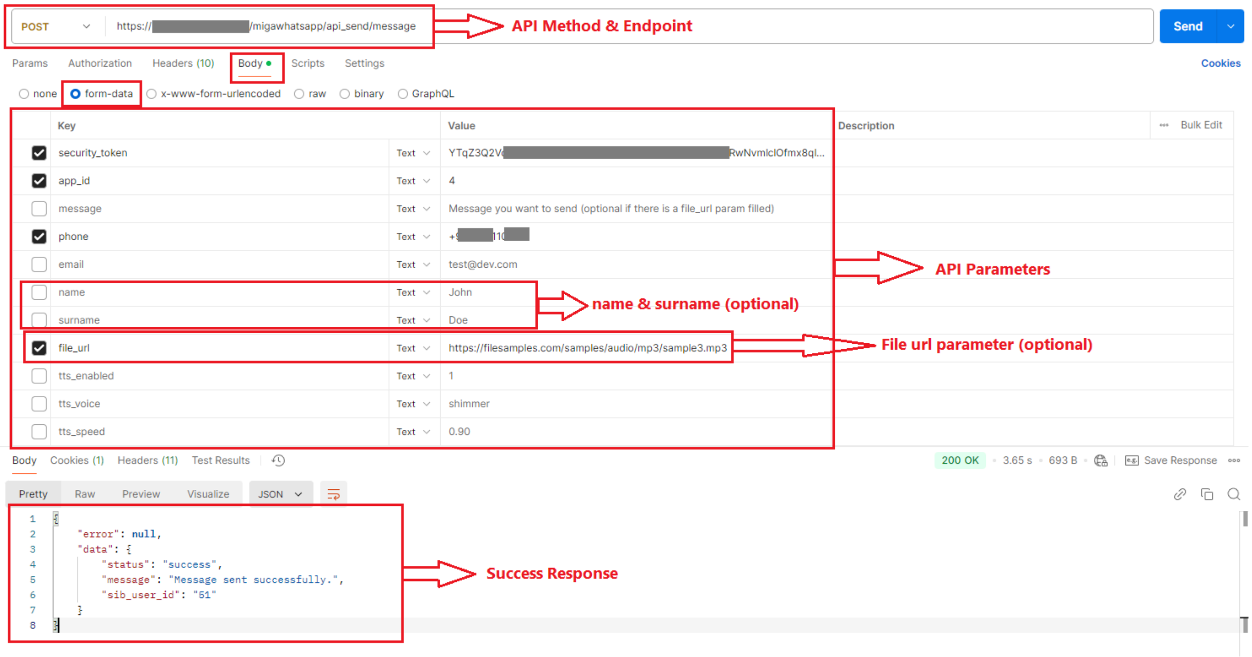
{
    "error": null,
    "data": {
        "status": "success",
        "message": "Message sent successfully.",
         "sib_user_id": SIBERIAN_USER_ID | 0,
        }
}
	The message was sent successfully.
	• `sib_user_id` can be `0` sometimes if the sent parameter was only phone and not an account found

{
    "error": null,
    "data": {
            "status": "success",
            "message": "Message sent successfully with the new user account creation",
            "sib_user_id": SIBERIAN_USER_ID,
            "sib_user_password": SIBERIAN_USER_PASSWROD,
        }
}
	The message was sent successfully with the new user account creation.
	• `sib_user_id` will always be there as a new account is created,
	• `sib_user_password` will also be sent in clear text form so that it can be shared with the created user

{
     "error": null,
     "data": {
         "groups": [
             {
                 "id": "123456789987452311@g.us",
                 "subject": "The Student Network"
             },
             {
                 "id": "123456789987452311@g.us",
                 "subject": "Humdum community "
             },
             ....
         ]
     }
}
	This is when calling the groups endpoint `/migawhatsapp/api_send/groups` , and in return, you get the list of group IDs and subject, which means the name of the group.

1.2.3. Example from POSTMAN [`message` API]:

The example from POSTMAN shows a successful API request to send a message. The attached image highlights important aspects of the request, indicating a problem-free execution.



POST https://[redacted]/migawhatsapp/api_send/message

API Method & Endpoint

	Key		Value	Description
☑	security_token	Text	YTqZ3Q2V[redacted]RwNvmlclOfmx8ql...	
☑	app_id	Text	4	
□	message	Text	Message you want to send (optional if there is a file_url param filled)	
☑	phone	Text	+[redacted]	
□	email	Text	test@dev.com	
□	name	Text	John	
□	surname	Text	Doe	
☑	file_url	Text	https://filesamples.com/samples/audio/mp3/sample3.mp3	
□	tts_enabled	Text	1	
□	tts_voice	Text	shimmer	
□	tts_speed	Text	0.90	

API Parameters

name & surname (optional)

File url parameter (optional)

```json
{
    "error": null,
    "data": {
        "status": "success",
        "message": "Message sent successfully.",
        "sib_user_id": "51"
    }
}
```

Success Response

1.2.4. Example with cURL (PHP) [`message` API]:

In the provided example, a code snippet is shown using CURL in PHP to successfully send a message. The code includes the necessary parameter values required for the API request to be executed without errors, ensuring the successful sending of the message.

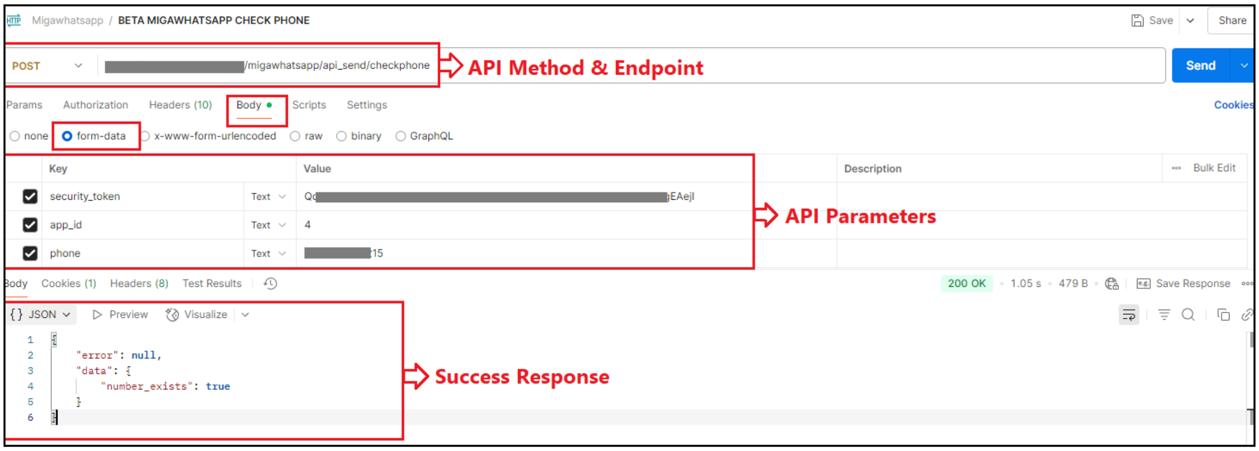
```php
<?php

$curl = curl_init();

curl_setopt_array($curl, [
  CURLOPT_URL => 'https://yourdomain.com/migawhatsapp/api_send/message',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_ENCODING => '',
  CURLOPT_MAXREDIRS => 10,
  CURLOPT_TIMEOUT => 0,
  CURLOPT_FOLLOWLOCATION => true,
  CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
  CURLOPT_CUSTOMREQUEST => 'POST',
  CURLOPT_POSTFIELDS => [
      'security_token' => some_token_here,
      'app_id' => '4',
      'phone' => '+39xxxxxxxxxx',
      'file_url' => 'https://ai.com/samples/audio/mp3/sample3.mp3'
  ],
  CURLOPT_HTTPHEADER => [
    'Cookie: editor=7ad7gciscoljijtnm988aania1'
  ],
]);

$response = curl_exec($curl);

curl_close($curl);
echo $response;
```

1.2.5. Sample API response using cURL (JSON) [`message` API]:

```
{
    "error": null,
    "data": {
        "status": "success",
        "message": "Message sent successfully with the new user account creation.",
        "sib_user_id": 22,
        "sib_user_password": "5oc3qwLwn"
    }
}
```

1.2.6. Example from POSTMAN [`checkphone` API]:

The example from POSTMAN shows a successful API request to check a phone number. The attached image highlights important aspects of the request, indicating a problem-free execution.



Migawhatsapp / BETA MIGAWHATSAPP CHECK PHONE

POST [redacted]/migawhatsapp/api_send/checkphone

API Method & Endpoint

○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

	Key		Value	Description
☑	security_token	Text	Q[redacted]EAejl	
☑	app_id	Text	4	
☑	phone	Text	[redacted]15	

API Parameters

```
{
    "error": null,
    "data": {
        "number_exists": true
    }
}
```

Success Response

1.2.7. Example with cURL (PHP) [checkphone API]:

In the provided example, a code snippet is shown using CURL in PHP to successfully check a phone number. The code includes the necessary parameter values required for the API request to be executed without errors, ensuring the successful checking of a phone.

```php
<?php

$curl = curl_init();

curl_setopt_array($curl, array(
  CURLOPT_URL => 'https://yourdomain.com/migawhatsapp/api_send/checkphone',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_ENCODING => '',
  CURLOPT_MAXREDIRS => 10,
  CURLOPT_TIMEOUT => 0,
  CURLOPT_FOLLOWLOCATION => true,
  CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
  CURLOPT_CUSTOMREQUEST => 'POST',
  CURLOPT_POSTFIELDS => [
      'security_token' => some_token_here,
      'app_id' => '4',
      'phone' => '+41223232322323'
  ],
  CURLOPT_HTTPHEADER => [
      'Cookie: editor=1g1e7jbg50bnc2sojsvbohq12l'
  ],
));
```

```
$response = curl_exec($curl);

curl_close($curl);
echo $response;
```

1.2.8. Sample API response using cURL (JSON) [checkphone API]:

```
{
    "error": null,
    "data": {
        "number_exists": true
    }
}
```

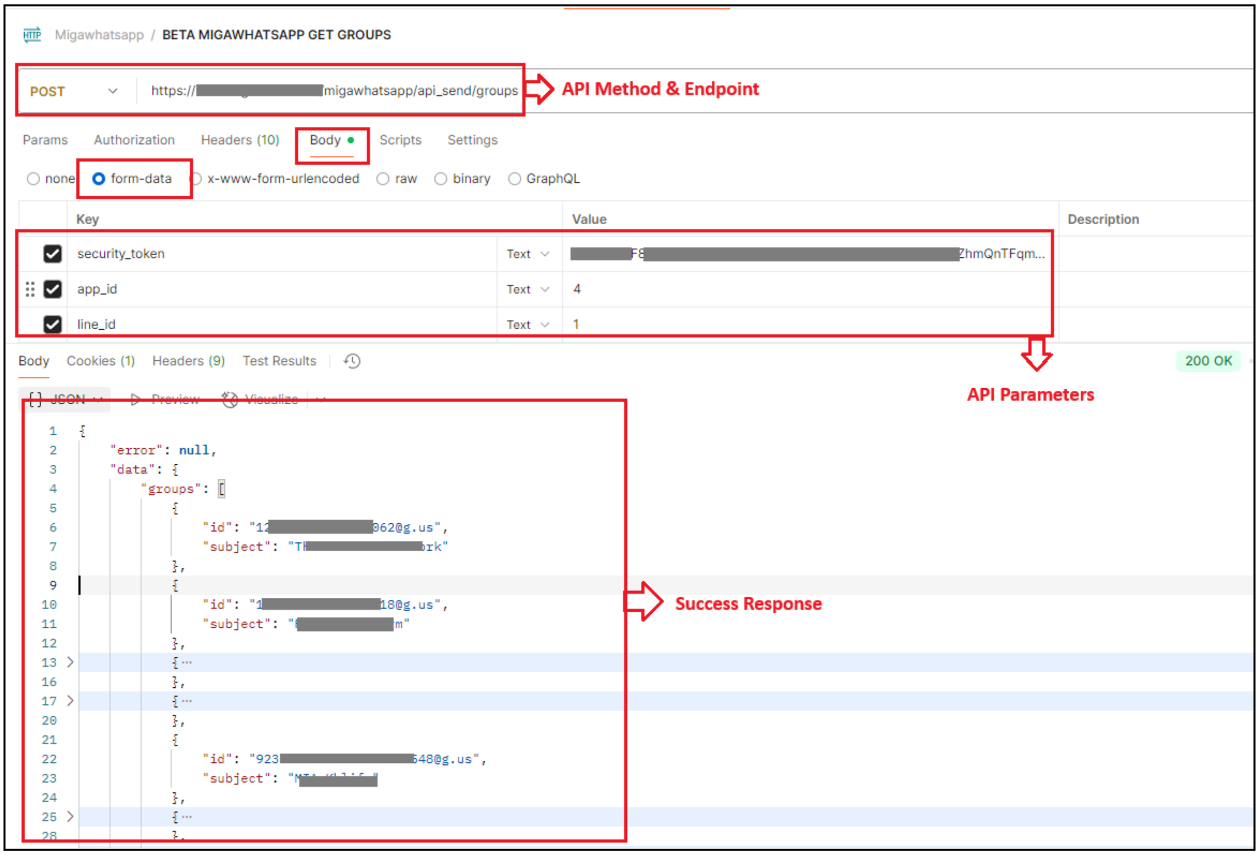
```
{
    "error": null,
    "data": {
        "number_exists": false
    }
}
```

[NEW IN 6.4.0]

```
{
    "error": "The service is currently unavailable due to an issue with the WA API Master. Please try again later.",
    "data": null
}
```

1.2.9. Example from POSTMAN [groups API]:

The example from POSTMAN shows a successful API request to get the list of groups. The attached image highlights important aspects of the request, indicating a problem-free execution.



Migawhatsapp / BETA MIGAWHATSAPP GET GROUPS

POST https://[redacted]migawhatsapp/api_send/groups

API Method & Endpoint

○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

	Key		Value	Description
☑	security_token	Text	[redacted]F8[redacted]ZhmQnTFqm...	
☑	app_id	Text	4	
☑	line_id	Text	1	

API Parameters

```
{
    "error": null,
    "data": {
        "groups": [
            {
                "id": "12[redacted]962@g.us",
                "subject": "T[redacted]rk"
            },
            {
                "id": "1[redacted]18@g.us",
                "subject": "[redacted]m"
            },
            {…
            },
            {…
            },
            {
                "id": "923[redacted]548@g.us",
                "subject": "[redacted]"
            },
            {…
            },
```

Success Response

1.2.10. Example with cURL (PHP) [groups API]:

In the provided example, a code snippet is shown using CURL in PHP to successfully load all the groups of a line. The code includes the necessary parameter values required for the API request to be executed without errors, ensuring the successful loading of groups.

```
<?php

$curl = curl_init();

curl_setopt_array($curl, array(
  CURLOPT_URL => 'https://yourdomain.com/migawhatsapp/api_send/groups',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_ENCODING => '',
  CURLOPT_MAXREDIRS => 10,
  CURLOPT_TIMEOUT => 0,
  CURLOPT_FOLLOWLOCATION => true,
  CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
  CURLOPT_CUSTOMREQUEST => 'POST',
  CURLOPT_POSTFIELDS => [
  'security_token' =>
'a1IZshwKF8ey319Si9WUF97PRGGvlEqUyEAKeixf6cRQpjAUSRFZhmQnTFqmWJz2',
  'app_id' => '4',
  'line_id' => '1'
  ],
  CURLOPT_HTTPHEADER => [
    'Cookie: editor=piqn5ic1c7e5s94vbr4vcda51a'
  ],
));

$response = curl_exec($curl);

curl_close($curl);
echo $response;
```

1.2.11. Sample API response using cURL (JSON) [`groups` API]:

```
{
    "error": null,
    "data": {
        "groups": [
            {
                "id": "123456789987452311@g.us",
                "subject": "The Student Network"
            },
            {
                "id": "123456789987452311@g.us",
                "subject": "Humdum community "
            },
            .........
        ]
    }
}
```

MIGAWHATSAPP CHAT HISTORY API Documentation

2. Introduction

This API class is designed to manage chat-related functionalities within the application. It handles various operations, such as initiating and managing chat sessions, validating user inputs, and ensuring proper sequencing of actions, such as requiring a start date before a start time. This class is responsible for maintaining the integrity and flow of chat interactions and providing necessary error messages and validations to ensure a smooth user experience. It integrates with other components of the application to facilitate seamless communication and data handling within chat sessions.

Now chat history API supports showing the media-related messages in the history JSON returned. You can find that from the same key where you were getting the message. The `message` key will have the media and message in case of a normal message, or both sometimes if both message and media exist. The message will be prepended to the media URL.

Additionally, you can now filter the chat history using the new `filter_line_id` `parameter`.

Chat history now has a parameter that will tell you IN & OUT, and I have also added a filter to filter by `in_out`.

2.1 API Endpoints

As access points to our API service, we have two endpoints:

2.1.1. Heartbeat Endpoint (`heartbeat`).

- `POST` Request URL:
  - `https://yourdomain.com/migawhatsapp/api_chat/heartbeat`

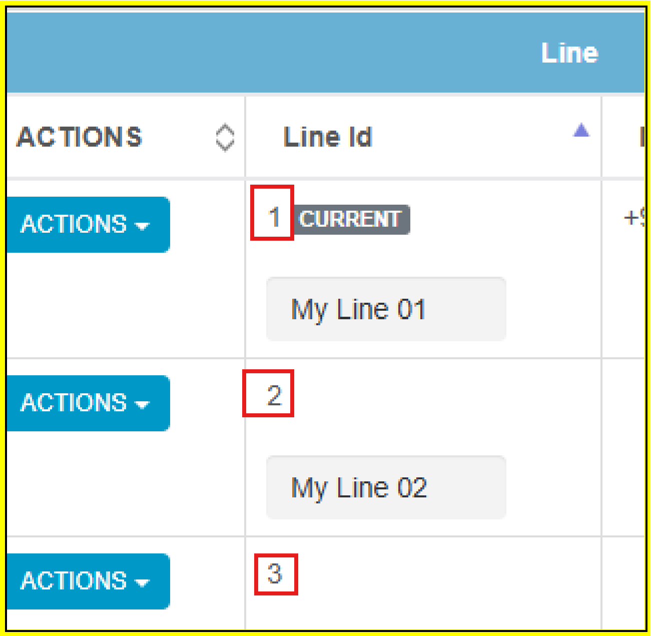
2.1.2. History Endpoint (`history`)

- `POST` Request URL:
  - `https://yourdomain.com/migawhatsapp/api_chat/history`

2.2 API Parameters

Both `heartbeat` and `history` endpoints use the same parameters, which are listed below:

Parameter	Type	Description
app_id	Int	**Mandatory** - YOUR_APP_ID (can be seen from the module's dashboard at the top left).
security_token	String	**Mandatory** - YOUR_SECURITY_TOKEN (this token is generated from the API Settings section under the module's Settings tab with the label Chat History API Token? Make sure you have enabled the chat history API as well.)
filter_history_id	Int	**Optional** - The history ID is the chat ID, and if you provide an ID for example, 10, then you will receive the history from 10 onward
filter_start_date	String (date)	**Optional** - The start date actually matches the `creatred_at` field of the record, and if you provide a date for example, in the valid format `yyyy-mm-dd`, then you will receive the history from that date onward
filter_start_time	String (time)	**Optional** - The start time matches the time for the `creatred_at` field of the record, and if you provide a `filter_start_date` and `filter_start_time,` for example in the valid format `HH:mm:ss`, then you will receive the history from that date and time onward. !!IMPORTANT!!: The time of your server will be used, so make sure to set up your application or server timezone properly
filter_phone_number	String (phone)	**Optional** - You can now filter the chat by a desired phone number. Make sure the phone number is in the correct format. Please add a phone number of 10-14 digits with + international country code at the beginning.
filter_origin	String	**Optional** - You can now filter the new API response with the following origin values: • WA API MASTER • WA API SLAVE • MIGACHAT • API • WEBHOOK
filter_destination	String	**Optional** - You can now filter the new API response with the following destination values: • WA API MASTER • WA API SLAVE • MIGACHAT

		- WEBHOOK - WA_USER
filter_email	String (email)	**Optional** - You can now filter the chat by a desired email address if the customer exists and has an email assigned. Make sure the email is valid in format.
filter_line_id	INT	**Optional** - You can now filter the chat by a `line_id` of the line that you can get from MAB under *Gateway Settings -> Whatsapp Line Management*:  Line ACTIONS | Line Id ACTIONS ▾ | 1 CURRENT My Line 01 ACTIONS ▾ | 2 My Line 02 ACTIONS ▾ | 3
filter_in_out	STRING	**Optional** - You can now filter the chat by an `in_out` parameter with possible values: - **IN** ○ All the incoming chats & messages - **OUT** ○ All the outgoing chats & messages

2.2.1. List of API calls' unsuccessful responses, and details:

Response	Description
`{"data":null,"error":"No parameter(s) were sent."}`	No required parameter is posted

`{"data":null,"error":"Security token missing."}`	Security token missing.
`{"data":null,"error":"App id missing."}`	App id missing.
`{"data":null,"error":"Invalid app id."}`	Invalid app id.
`{"data":null,"error ":"API is disabled."}`	API is disabled (from the settings).
`{"data":null,"error ":"Security token missing."}`	Security token missing.
`{"data":null,"error ":"Security token mismatched."}`	Security token mismatch (the wrong token is used).
`{"data":null,"error ":"This app has not yet configured the gateway settings so you cannot get the history."}`	This app has not yet configured the gateway settings, so you cannot get the history."
`{"data":null,"error ":"Invalid date format. Please provide the date in yyyy-mm-dd format."}`	Invalid date format. Please provide the date in `yyyy-mm-dd` format. This will happen if you use the `filter_start_date filter` and the date format is not correct.
`{"data":null,"error ":"Invalid time format. Please provide the time in HH:mm:ss format."}`	Invalid time format. Please provide the time in `HH:mm:ss` format. This will happen if you use the `filter_start_time` filter and the incorrect time format.
`{"data":null,"error ":"Please provide the start date before providing the start time."}`	Please provide the start date before providing the start time. This error means that you have sent the filter_start_time filter value only without the filter_start_date value.
`{"data":null,"error ":"The phone number is not correct. Please add a phone between 10-14 digits with + international country code at the beginning."}`	This error means that you have sent the `filter_phone_number` filter value without following the international phone number practices.
`{"data":null,"error ":"Invalid origin value. Please provide a valid origin value."}`	This error means that you have sent the `filter_origin` filter value, but the value is not inside the possible list of values. Which are: ● WA API MASTER ● WA API SLAVE

	● MIGACHAT ● API ● MAB
`{"data":null,"error ":"Invalid destination value. Please provide a valid destination value."}`	This error means that you have sent the `filter_destination` filter value, but the value is not inside the possible list of values. Which are: ● WA API MASTER ● WA API SLAVE ● MIGACHAT ● WEBHOOK
`{"data":null,"error ":"Invalid email address. Please provide a valid email address."}`	This error means that you have sent the `filter_email` filter value without following the best practices for a valid email address.
`{"data":null,"error ":"Invalid line ID. Please provide a valid line ID."}`	This error indicates that you have sent the `filter_line_id` filter value without adhering to the best practices for a numeric value.
`{"data":null,"error ":"Line id not found. Please provide a valid line id."}`	This error indicates that you have sent the `filter_line_id` filter value, and it does not exist for that app.
`{"data":null,"error ":"Invalid in/out value. Please provide a valid in/out value."}`	This error indicates that you have sent the `filter_in_out` filter value, and the value is not valid. The possible values are: ● IN ● OUT

2.2.2. List of API calls' successful responses and details :

Response	Description

`{`
`"error": null,`
`"data": {`
`"status": "success",`
`"message": "No chat history found."`
	No chat history was found.

```
    }
}
```

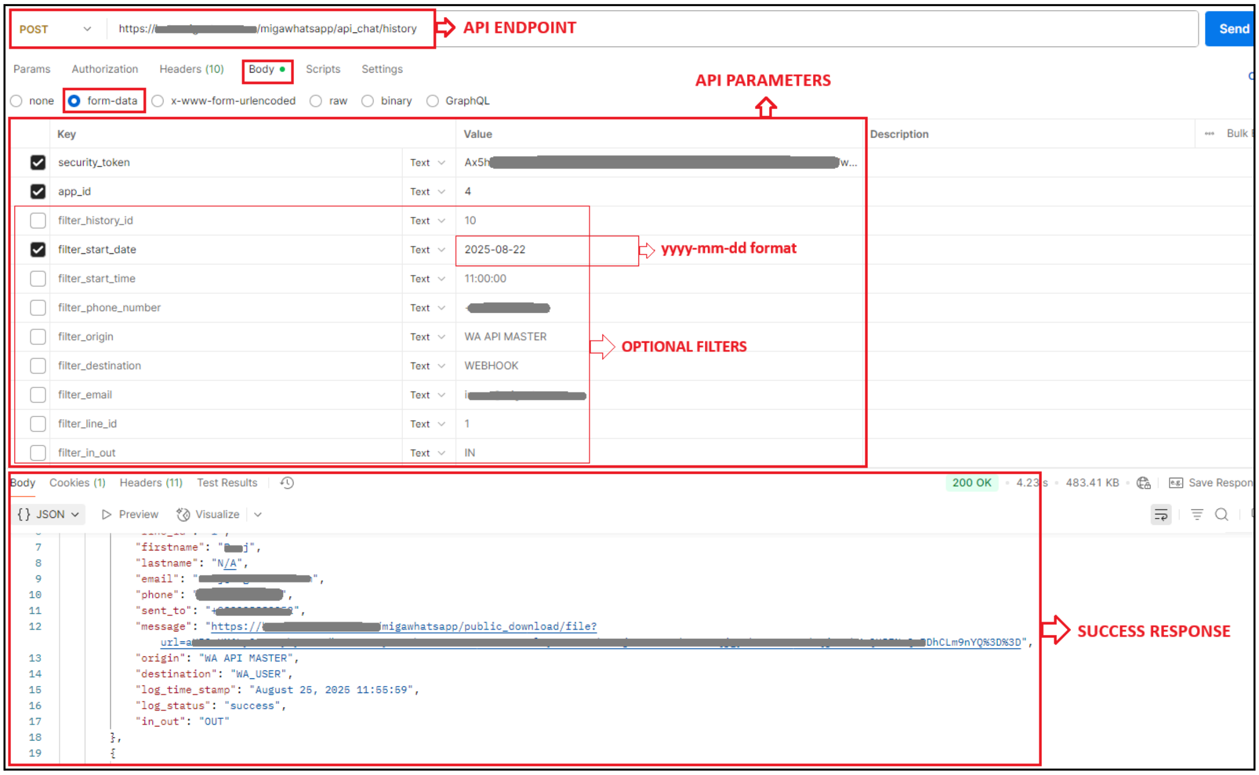
```
{
    "error": null,
    "data": [
        {
            "log_id": "56",
            "line_id": "2",
            "firstname": "John | N/A",
            "lastname": "Doe | N/A",
            "email": "john@mail.com | N/A",
            "phone": "+9200000000 | N/A",
            "sent_to": null | "+9200000000",
            "origin": "WA API MASTER | API |...",
            "destination": "WA API MASTER | WEB…",
            "message": "Some Message | N/A",
            "log_time_stamp": "February 11, 2025
07:47:10",
            "log_status": "success | error |
NO_COUNTRY…",
            "in_out": "IN | OUT",
        },
        { …
        },
}
```

The final success response with history will resemble the following JSON. It will have fields such as:

- log_id
- line_id
- firstname
- lastname
- email
- phone
- sent_to [If you see something in the 'Sent To' field other than null, that means this is the message or chat sent from the owner to someone, and the 'Sent To' field holds the receiver's phone number]
- message
- origin
- destination
- log_time_stamp
- log_status
- in_out

2.2.3. Example from POSTMAN:

The example from POSTMAN shows a successful API request to get history. The attached image highlights important aspects of the request, indicating a problem-free execution.



POST https://[redacted]/migawhatsapp/api_chat/history

API ENDPOINT

○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

API PARAMETERS

	Key		Value
☑	security_token	Text	Ax5h[redacted]w...
☑	app_id	Text	4
□	filter_history_id	Text	10
☑	filter_start_date	Text	2025-08-22
□	filter_start_time	Text	11:00:00
□	filter_phone_number	Text	[redacted]
□	filter_origin	Text	WA API MASTER
□	filter_destination	Text	WEBHOOK
□	filter_email	Text	i[redacted]
□	filter_line_id	Text	1
□	filter_in_out	Text	IN

yyyy-mm-dd format

OPTIONAL FILTERS

```
            "firstname": "[redacted]j",
            "lastname": "N/A",
            "email": "[redacted]",
            "phone": "[redacted]",
            "sent_to": "+[redacted]",
            "message": "https://[redacted]migawhatsapp/public_download/file?
              url=a[redacted]DhCLm9nYQ%3D%3D",
            "origin": "WA API MASTER",
            "destination": "WA_USER",
            "log_time_stamp": "August 25, 2025 11:55:59",
            "log_status": "success",
            "in_out": "OUT"
        },
        {
```

SUCCESS RESPONSE

2.2.4. Example with cURL (PHP):

In the provided example, a code snippet is shown using CURL in PHP to successfully load the chat history. The code includes the necessary parameter values required for the API request to be executed without errors, ensuring the successful loading of history.

```
$curl = curl_init();

curl_setopt_array($curl, array(
  CURLOPT_URL => 'https://yourdomain.com/migawhatsapp/api_chat/history',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_ENCODING => '',
  CURLOPT_MAXREDIRS => 10,
  CURLOPT_TIMEOUT => 0,
  CURLOPT_FOLLOWLOCATION => true,
  CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
  CURLOPT_CUSTOMREQUEST => 'POST',
  CURLOPT_POSTFIELDS => array('security_token' => 'asdsadsa323232saadasds','app_id' =>
'4','filter_history_id' => '10','filter_start_date' => '2024-09-21'),
  CURLOPT_HTTPHEADER => array(
    'Cookie: editor=944ri3rvm9egcgi3s8jqi4flok'
  ),
));
```

```
$response = curl_exec($curl);

curl_close($curl);
echo $response;
```

2.2.5. Sample API response using cURL (JSON):

```
{
    "error": null,
    "data": [
        {
            "log_id": "56",
            "line_id": "2",
            "firstname": "John | N/A",
            "lastname": "Doe | N/A",
            "email": "john@mail.com | N/A",
            "phone": "+9200000000 | N/A",
            "sent_to": null | "+9200000000",
            "origin": "WA API MASTER | API |...",
            "destination": "WA API MASTER | WEB…",
            "message": "Some Message | N/A",
            "log_time_stamp": "February 11, 2025 07:47:10",
            "log_status": "success | error | NO_COUNTRY…",
            "in_out": "IN | OUT",
        },
        { …
        },
}
```