

# Scheduled Messages

## Introduction

The **scheduled messages** feature allows users to select a date and time in the future for messages to be sent. Users can schedule messages starting from 5 minutes to 30 days in advance. Scheduling a message can be useful when an important message must be sent at a later time. This feature isn't supported in Supergroup channels or open channels.

## Expected benefits

### Flexibility in time management

Scheduled messages enables users to manage their time flexibly by scheduling messages to be sent at a later time. For example, if a user wants to write a message now but send at a later time, the user is able to do so by scheduling a message. This way, users don't have to wait until the right time, and can avoid forgetting to send an important message.

### Various use cases to remind and trigger clients

Scheduled messages can also serve as a way to nudge various actions. For instance, if an admin wants to plan an announcement in advance, they can schedule admin messages to send out the announcement to members. It can also work as a reminder for users to take certain actions before the deadline (e.g. fill out a form within 3 days).

## Limitations

### Beta feature

Note that this is a beta feature, so properties and flows may be changed in the future to support better user experience. Having said that, we welcome any feedback and requests for improvement, so please reach out to your Sendbird representatives to connect with us.

---

## Group channel only

Scheduled message is only supported in [group channels](#) and not in any other channels (open or Supergroup) for the time being.

## Rate limits

Note that there is a limit of 100 scheduled messages that can be sent per user, and 1000 scheduled messages for admin messages.

If a user has reached their limit of 100 scheduled messages, the following messages after the limit will not be sent. Currently, it returns a 429 - Too Many Request error. Furthermore, there's a chance that when a message is actually dispatched to recipients at the scheduled time, it might hit the general rate limit for sending messages and therefore not be sent. This will also trigger 429 - Too Many Request error as well.

---

## API specification

### Scheduled message status

Status	Description
pending	Indicates that a message has been scheduled and is waiting to be sent at the scheduled time.
in_queue	Indicates that a cron job is processing this message.
sent	Indicates that a message was successfully sent.
failed	Indicates that a scheduled message was not successfully sent.
canceled	Indicates that a scheduled message was manually canceled and removed from the schedule before it was sent.

### Resource representation

The following tables show the list of properties in a scheduled text message, file message, and admin message resources.

List of properties for creating a scheduled text message

Property name	Type	Description
scheduled_message_id	long	The unique ID of the scheduled message.
type	string	The type of the message. The value is <code>MESG</code> for a text message.
message	string	The content of the message.

channel_url	string	The unique URL of the channel where the message is sent to.
sender_id	string	The user ID of the sender.
scheduled_status	string	The status of a scheduled message. Valid values are <code>pending</code> , <code>in_queue</code> , <code>sent</code> , <code>failed</code> , and <code>canceled</code> .
scheduled_at	long	The time to send the message, in Unix milliseconds format. Since messages are scheduled in minutes, values less than seconds are discarded. The term between current time and <code>scheduled_at</code> should be between 5 minutes and 30 days.
created_at	long	The time when the message was sent in Unix milliseconds format.
sent_at	long	The sent time of the scheduled message in Unix milliseconds format. If it's not sent yet, the value is 0.
custom_type	string	Specifies a custom message type used for message grouping. The length is limited to 128 characters.
data	string	Specifies additional message information such as custom font size, font type or <code>JSON</code> formatted string.
send_push	boolean	Determines whether to send a push notification of the message to the channel members. This property only applies to group channels. (Default: true)
mention_type	string	Specifies the mentioning method that indicates which user receives a notification of the message. Acceptable values are <code>users</code> and <code>channels</code> . If set to <code>users</code> , only users specified in the <code>mentioned_user_ids</code> property below are notified. If set to <code>channels</code> , all users in the channel are notified. (Default: users)
mentioned_user_ids[ ]	array of strings	Specifies an array of one or more IDs of the users who will receive a notification of the message.
is_silent	boolean	Determines whether to send a message without

		<p>updating some of the following channel properties. If set to true, the channel's last_message is updated only for the sender while its unread_message_count remains unchanged for all channel members. Also, a push notification isn't sent to the users receiving the message. If the message is sent to a hidden channel, the channel remains hidden. (Default: false)</p> <p>* Once the value of this property is set, it's irreversible.</p>
mark_as_read	boolean	Determines whether to mark the message as read for the sender. If set to false, then the sender's unread_count and read_receipt remain unchanged after the message is sent. (Default: true)
sorted_metaarray	array of objects	Specifies a JSON object of one or more key-values items which store additional message information. Each item consists of a key and the values in an array. Items are saved and will be returned in the exact order they've been specified.
dedup_id	string	Specifies the unique message ID created by another system. In general, this property is used to prevent the same message data from getting inserted when migrating messages from another system to Sendbird server. If specified, the server performs a duplicate check using the property value.
apns_bundle_id	string	Specifies the bundle ID of the client app in order to send a push notification to iOS devices. You can find this in Settings > Chat > Notifications > Push notification services on the Sendbird Dashboard.
apple_critical_alert_options	object	Specifies options that support <a href="#">Apple's critical alerts</a> and checks whether the message is a critical alert.
sound	string	Specifies the name of a sound file that is used for critical alerts.
volume	float	Specifies the volume of the critical alert sound. The volume ranges from 0.0 to 1.0, which indicates silent and full volume, respectively. (Default: 1.0)

List of properties for creating a scheduled file message

Property name	Type	Description
scheduled_message_id	long	The unique ID of the scheduled message.
type	string	The type of the message. The value is <code>MESG</code> for a text message.
file	string	Depending on the file upload method, this specifies the data of the file to upload to Sendbird server in raw binary format, or the location of the file. When sending a request containing a file, you must change the value of the content-type header to <code>multipart/form-data</code> ; <code>boundary={your_unique_boundary_string}</code> in the request. If this file property is specified, the <code>url</code> property is not required.  * This doesn't allow a converted <code>base64-encoded</code> string from a file as its value.
file.url	string	Specifies the URL of the file hosted on the server of your own or other third-party companies. If this <code>url</code> property is specified, the <code>file</code> property is not required.
file_name	string	If the <code>file</code> property is used for file upload, this property is automatically specified by Sendbird server. Otherwise, specify the name of the file directly.
file_size	int	If the <code>file</code> property is used for file upload, this property is automatically specified by Sendbird server. Otherwise, specify the size of the file directly.
file_type	string	If the <code>file</code> property is used for file upload, this property is automatically specified by Sendbird server. Otherwise, specify the media type of the file directly.
thumbnails[]	array of strings	Specifies an array of one or more URLs of external thumbnail images that are generated from the image specified by the <code>url</code> property.

channel_url	string	The unique URL of the channel where the message is sent to.
sender_id	string	The user ID of the sender.
scheduled_status	string	The status of a scheduled message. Valid values are <code>pending</code> , <code>in_queue</code> , <code>sent</code> , <code>failed</code> , and <code>canceled</code> .
scheduled_at	long	The time to send the message, in Unix milliseconds format. Since messages are scheduled in minutes, values less than seconds are discarded. The term between current time and <code>scheduled_at</code> should be between 5 minutes and 30 days.
created_at	long	The time when the message was sent in Unix milliseconds format.
sent_at	long	The sent time of the scheduled message in Unix milliseconds format. If it's not sent yet, the value is 0.
custom_type	string	Specifies a custom message type used for message grouping. The length is limited to 128 characters.
data	string	Specifies additional message information such as custom font size, font type or <code>JSON</code> formatted string.
require_auth	boolean	<p>Determines whether the files and thumbnail images in a message are only accessible by users who are in the same channel. This property is only effective when using the share encrypted files feature. This feature encrypts all types of uploaded files and the automatically-generated thumbnail images with <code>AES256</code>, stores them securely in Sendbird server, and enables only the users with access to share the files. You can use this property to restrict access to the files.</p> <p>If set to <code>true</code>, the files are uploaded first, then encrypted, and finally stored along with the thumbnail images which are encrypted as well. When accessing them, the parameter <code>auth</code> should be added to the end</p>

		of the file URLs. The parameter <code>auth</code> is specified with an encryption key of the user such as <code>?auth=RW5b2RIHaMdGV4eA==</code> . An encryption key managed internally by the Sendbird server is generated every time a user logs into Sendbird server and is valid for three days starting from the last login. The system mainly uses the key to check if a user is properly accessing the data of the application. (Default: true)
<code>send_push</code>	boolean	Determines whether to send a push notification of the message to the channel members. This property only applies to group channels. (Default: true)
<code>mention_type</code>	string	Specifies the mentioning method that indicates which user receives a notification of the message. Acceptable values are <code>users</code> and <code>channels</code> . If set to <code>users</code> , only users specified in the <code>mentioned_user_ids</code> property below are notified. If set to <code>channels</code> , all users in the channel are notified. (Default: <code>users</code> )
<code>mentioned_user_ids[ ]</code>	array of objects	Specifies an array of one or more IDs of the users who will receive a notification of the message.
<code>is_silent</code>	boolean	Determines whether to send a message without updating some of the following channel properties. If set to <code>true</code> , the channel's <code>last_message</code> is updated only for the sender while its <code>unread_message_count</code> remains unchanged for all channel members. Also, a push notification isn't sent to the users receiving the message. If the message is sent to a hidden channel, the channel remains hidden. (Default: <code>false</code> )  * Once the value of this property is set, it's irreversible.
<code>mark_as_read</code>	boolean	Determines whether to mark the message as read for the sender. If set to <code>false</code> , then the sender's <code>unread_count</code> and <code>read_receipt</code> remain unchanged after the message is sent. (Default: <code>true</code> )
<code>sorted_metaarray</code>	array of objects	Specifies a <code>JSON</code> object of one or more key-values items which store additional message information. Each item consists of a key and the values in an array. Items are saved and will be returned in the exact order they've been specified.
<code>dedup_id</code>	string	Specifies the unique message ID created by another system. In general, this property is used to prevent the



		same message data from getting inserted when migrating messages from another system to Sendbird server. If specified, the server performs a duplicate check using the property value.
apns_bundle_id	string	Specifies the bundle ID of the client app in order to send a push notification to iOS devices. You can find this in Settings > Chat > Notifications > Push notification services on the Sendbird Dashboard.
apple_critical_alert_options	object	Specifies options that support <a href="#">Apple's critical alerts</a> and checks whether the message is a critical alert.
sound	string	Specifies the name of a sound file that is used for critical alerts.
volume	float	Specifies the volume of the critical alert sound. The volume ranges from 0.0 to 1.0, which indicates silent and full volume, respectively. (Default: 1.0)

#### List of properties for creating a scheduled admin message

Property name	Type	Description
scheduled_message_id	long	The unique ID of the scheduled message.
type	string	The type of the message. The value is MESSG for a text message.
message	string	The content of the message.
channel_url	string	The unique URL of the channel where the message is sent to.
sender_id	string	The user ID of the sender.

scheduled_status	string	The status of a scheduled message. Valid values are pending, in_queue, sent, failed, and canceled.
scheduled_at	long	The time to send the message, in Unix milliseconds format. Since messages are scheduled in minutes, values less than seconds are discarded. The term between current time and scheduled_at should be between 5 minutes and 30 days.
created_at	long	The time when the message was sent in Unix milliseconds format.
sent_at	long	The sent time of the scheduled message in Unix milliseconds format. If it's not sent yet, the value is 0.
custom_type	string	Specifies a custom message type used for message grouping. The length is limited to 128 characters.
data	string	Specifies additional message information such as custom font size, font type or JSON formatted string.
send_push	boolean	Determines whether to send a push notification of the message to the channel members. This property only applies to group channels. (Default: true)
mention_type	string	Specifies the mentioning method that indicates which user receives a notification of the message. Acceptable values are users and channels. If set to users, only users specified in the mentioned_user_ids property below are notified. If set to channels, all users in the channel are notified. (Default: users)
mentioned_user_ids[ ]	array of strings	Specifies an array of one or more IDs of the users who will receive a notification of the message.
is_silent	boolean	Determines whether to send a message without updating some of the following channel properties. If set to true, the channel's last_message is updated only for the sender while its unread_message_count remains unchanged for all channel members. Also, a push notification isn't sent to the users receiving the message. If the message is sent to a hidden channel, the channel remains hidden. (Default: false)

		* Once the value of this property is set, it's irreversible.
sorted_metaarray	array of objects	Specifies a JSON object of one or more key-values items which store additional message information. Each item consists of a key and the values in an array. Items are saved and will be returned in the exact order they've been specified.
dedup_id	string	Specifies the unique message ID created by another system. In general, this property is used to prevent the same message data from getting inserted when migrating messages from another system to Sendbird server. If specified, the server performs a duplicate check using the property value.
apns_bundle_id	string	Specifies the bundle ID of the client app in order to send a push notification to iOS devices. You can find this in Settings > Chat > Notifications > Push notification services on the Sendbird Dashboard.
apple_critical_alert_options	object	Specifies options that support <a href="#">Apple's critical alerts</a> and checks whether the message is a critical alert.

## Actions

- API endpoints are relative to the base URL allocated to your Sendbird application. The base URL for the following endpoints refers to `https://api-{application_id}.sendbird.com/v3`.

| **Note:** If you want to know the ID and base URL of your application, sign in to [Sendbird Dashboard](#), go to **Settings > Application > General**, and check the **Application ID** and **API request URL**.

---

## Create a scheduled message

Creates a new scheduled message. If a user leaves the channel before the scheduled time for the message to be sent, the scheduled message will be removed.

### HTTP request

```
POST
https:api-{application_id}.sendbird.com/v3/group_channels/{channel_url}/scheduled_messages
```

### Parameters

The following table lists the parameters that this action supports.

Required parameter	Type	Description
channel_url	string	Specifies the URL of the channel.

### Request body

The following table lists the properties of an HTTP request that this action supports for creating a scheduled text message, file message, and admin message.

List of properties for creating a scheduled text message

Required	Type	Description
message_type	string	Specifies the type of the message. The value of <code>MESG</code> represents a text message.
user_id	string	Specifies the user ID of the sender.
message	string	Specifies the content of the message.

scheduled_at	long	The time to send the message, in Unix milliseconds format. Since messages are scheduled in minutes, values less than seconds are discarded. The term between current time and <code>scheduled_at</code> should be between 5 minutes and 30 days.
--------------	------	--

Optional	Type	Description
custom_type	string	Specifies a custom message type used for message grouping. The length is limited to 128 characters.
data	string	Specifies additional message information such as custom font size, font type or JSON formatted string.
send_push	boolean	Determines whether to send a push notification of the message to the channel members. This property only applies to group channels. (Default: true)
mention_type	string	Specifies the mentioning method that indicates which user receives a notification of the message. Acceptable values are <code>users</code> and <code>channels</code> . If set to <code>users</code> , only users specified in the <code>mentioned_user_ids</code> property below are notified. If set to <code>channels</code> , all users in the channel are notified. (Default: <code>users</code> )
mentioned_user_ids[ ]	array of strings	Specifies an array of one or more IDs of the users who will receive a notification of the message.
is_silent	boolean	Determines whether to send a message without updating some of the following channel properties. If set to <code>true</code> , the channel's <code>last_message</code> is updated only for the sender while its <code>unread_message_count</code> remains unchanged for all channel members. Also, a push notification isn't sent to the users receiving the message. If the message is sent to a hidden channel, the channel remains hidden. (Default: <code>false</code> )  * Once the value of this property is set, it's irreversible.
mark_as_read	boolean	Determines whether to mark the message as read for the sender. If set to <code>false</code> , then the sender's <code>unread_count</code> and <code>read_receipt</code> remain unchanged after the message is sent. (Default: <code>true</code> )

sorted_metaarray	array of objects	Specifies a <b>JSON</b> object of one or more key-values items which store additional message information. Each item consists of a key and the values in an array. Items are saved and will be returned in the exact order they've been specified.
dedup_id	string	Specifies the unique message ID created by another system. In general, this property is used to prevent the same message data from getting inserted when migrating messages from another system to Sendbird server. If specified, the server performs a duplicate check using the property value.
apns_bundle_id	string	Specifies the bundle ID of the client app in order to send a push notification to iOS devices. You can find this in <b>Settings &gt; Chat &gt; Notifications &gt; Push notification services</b> on the Sendbird Dashboard.
apple_critical_alert_options	object	Specifies options that support <a href="#">Apple's critical alerts</a> and checks whether the message is a critical alert.
sound	string	Specifies the name of a sound file that is used for critical alerts.
volume	float	Specifies the volume of the critical alert sound. The volume ranges from 0.0 to 1.0, which indicates silent and full volume, respectively. (Default: 1.0)

#### List of properties for creating a scheduled file message

Required	Type	Description
message_type	string	Specifies the type of the message. The value of <b>FILE</b> represents a file message.
user_id	string	Specifies the user ID of the sender.
file	string	Depending on the file upload method, this specifies the data of the file to upload to Sendbird server in raw binary format, or the location of the file. When sending a request containing a file, you must change the value of the content-type header to multipart/form-data; boundary={your_unique_boundary_string} in the

		<p>request. If this <code>file</code> property is specified, the <code>url</code> property is not required.</p> <p>* This doesn't allow a converted <code>base64-encoded</code> string from a file as its value.</p>
<code>url</code>	string	Specifies the URL of the file which is hosted on the server of your own or other external third party companies. If this <code>url</code> property is specified, the <code>file</code> property is not required.
<code>scheduled_at</code>	long	The time to send the message, in Unix milliseconds format. Since messages are scheduled in minutes, values less than seconds are discarded. The term between current time and <code>scheduled_at</code> should be between 5 minutes and 30 days.

Optional	Type	Description
<code>file_name</code>	string	If the <code>file</code> property is used for file upload, this property is automatically specified by Sendbird server. Otherwise, specify the name of the file directly.
<code>file_size</code>	int	If the <code>file</code> property is used for file upload, this property is automatically specified by Sendbird server. Otherwise, specify the size of the file directly.
<code>file_type</code>	string	If the <code>file</code> property is used for file upload, this property is automatically specified by Sendbird server. Otherwise, specify the media type of the file directly.
<code>thumbnails[]</code>	array of strings	Specifies an array of one or more URLs of external thumbnail images that are generated from the image specified by the <code>url</code> property.
<code>custom_type</code>	string	Specifies a custom message type used for message grouping. The length is limited to 128 characters.
<code>data</code>	string	Specifies additional message information such as custom font size, font type or <code>JSON</code> formatted string.
<code>require_auth</code>	boolean	Determines whether the files and thumbnail images in a message are only accessible by users who are in the same channel. This property is only effective when

		<p>using the share encrypted files feature. This feature encrypts all types of uploaded files and the automatically-generated thumbnail images with AES256, stores them securely in Sendbird server, and enables only the users with access to share the files. You can use this property to restrict access to the files.</p> <p>If set to true, the files are uploaded first, then encrypted, and finally stored along with the thumbnail images which are encrypted as well. When accessing them, the parameter auth should be added to the end of the file URLs. The parameter auth is specified with an encryption key of the user such as ?auth=RW5b2RIHaMdGV4eA==. An encryption key managed internally by the Sendbird server is generated every time a user logs into Sendbird server and is valid for three days starting from the last login. The system mainly uses the key to check if a user is properly accessing the data of the application. (Default: true)</p>
send_push	boolean	Determines whether to send a push notification of the message to the channel members. This property only applies to group channels. (Default: true)
mention_type	string	Specifies the mentioning method that indicates which user receives a notification of the message. Acceptable values are users and channels. If set to users, only users specified in the mentioned_user_ids property below are notified. If set to channels, all users in the channel are notified. (Default: users)
mentioned_user_ids[ ]	array of strings	Specifies an array of one or more IDs of the users who will receive a notification of the message.
is_silent	boolean	<p>Determines whether to send a message without updating some of the following channel properties. If set to true, the channel's last_message is updated only for the sender while its unread_message_count remains unchanged for all channel members. Also, a push notification isn't sent to the users receiving the message. If the message is sent to a hidden channel, the channel remains hidden. (Default: false)</p> <p>* Once the value of this property is set, it's irreversible.</p>
mark_as_read	boolean	Determines whether to mark the message as read for the sender. If set to false, then the sender's



		unread_count and read_receipt remain unchanged after the message is sent. (Default: true)
sorted_metaarray	array of objects	Specifies a JSON object of one or more key-values items which store additional message information. Each item consists of a key and the values in an array. Items are saved and will be returned in the exact order they've been specified.
dedup_id	string	Specifies the unique message ID created by another system. In general, this property is used to prevent the same message data from getting inserted when migrating messages from another system to Sendbird server. If specified, the server performs a duplicate check using the property value.
apns_bundle_id	string	Specifies the bundle ID of the client app in order to send a push notification to iOS devices. You can find this in Settings > Chat > Notifications > Push notification services on the Sendbird Dashboard.
apple_critical_alert_options	object	Specifies options that support <a href="#">Apple's critical alerts</a> and checks whether the message is a critical alert.
sound	string	Specifies the name of a sound file that is used for critical alerts.
volume	float	Specifies the volume of the critical alert sound. The volume ranges from 0.0 to 1.0, which indicates silent and full volume, respectively. (Default: 1.0)

#### List of properties for creating a scheduled admin message

Required	Type	Description
message_type	string	Specifies the type of the message. The value of ADMM represents an admin message.
user_id	string	Specifies the user ID of the sender.
message	string	Specifies the content of the message.
scheduled_at	long	The time to send the message, in Unix milliseconds format. Since messages are scheduled in minutes, values less than seconds are discarded. The term

		between current time and <code>scheduled_at</code> should be between 5 minutes and 30 days.
--	--	---

Optional	Type	Description
<code>custom_type</code>	string	Specifies a custom message type used for message grouping. The length is limited to 128 characters.
<code>data</code>	string	Specifies additional message information such as custom font size, font type or <code>JSON</code> formatted string.
<code>send_push</code>	boolean	Determines whether to send a push notification of the message to the channel members. This property only applies to group channels. (Default: true)
<code>mention_type</code>	string	Specifies the mentioning method that indicates which user receives a notification of the message. Acceptable values are <code>users</code> and <code>channels</code> . If set to <code>users</code> , only users specified in the <code>mentioned_user_ids</code> property below are notified. If set to <code>channels</code> , all users in the channel are notified. (Default: <code>users</code> )
<code>mentioned_user_ids[ ]</code>	array of strings	Specifies an array of one or more IDs of the users who will receive a notification of the message.
<code>is_silent</code>	boolean	Determines whether to send a message without updating some of the following channel properties. If set to true, the channel's <code>last_message</code> is updated only for the sender while its <code>unread_message_count</code> remains unchanged for all channel members. Also, a push notification isn't sent to the users receiving the message. If the message is sent to a hidden channel, the channel remains hidden. (Default: false)  * Once the value of this property is set, it's irreversible.
<code>sorted_metaarray</code>	array of objects	Specifies a <code>JSON</code> object of one or more key-values items which store additional message information. Each item consists of a key and the values in an array. Items are saved and will be returned in the exact order they've been specified.
<code>dedup_id</code>	string	Specifies the unique message ID created by another

		system. In general, this property is used to prevent the same message data from getting inserted when migrating messages from another system to Sendbird server. If specified, the server performs a duplicate check using the property value.
apns_bundle_id	string	Specifies the bundle ID of the client app in order to send a push notification to iOS devices. You can find this in <b>Settings &gt; Chat &gt; Notifications &gt; Push notification services</b> on the Sendbird Dashboard.
apple_critical_alert_options	object	Specifies options that support <a href="#">Apple's critical alerts</a> and checks whether the message is a critical alert.

## Response

If successful, this action returns a scheduled message resource in the response body.

In the case of an error, an error object is returned. A detailed list of error codes is available in our [documentation](#).

## View a scheduled message

Retrieves information on a specific scheduled message.

## HTTP request

```
GET
https://api-{application_id}.sendbird.com/v3/group_channels/{channel_url}/scheduled_messages/{scheduled_message_id}
```

## Parameters

Required	Type	Description
----------	------	-------------

channel_url	string	Specifies the URL of the channel.
scheduled_message_id	long	Specifies the unique ID of the scheduled message.

## Response

If successful, this action returns a scheduled message resource in the response body.

In the case of an error, an error object is returned. A detailed list of error codes is available in our [documentation](#).

## List scheduled messages

Retrieves a list of scheduled messages. A user can only see the list of their own scheduled messages.

## HTTP request

```
GET
https://api-{application_id}.sendbird.com/v3/scheduled_messages
```

## Parameters

The following table lists the parameters that this action supports.

Optional	Type	Description
channel_url	string	Specifies the URL of the channel.
channel_type	string	Specifies the channel type of channel_url. * Only group_channels is supported as of now.
sender_id	string	Specifies the user ID of the sender.

token	string	Specifies a token that indicates the starting index of a chunk of results to retrieve. If not specified, the index is set as 0.
limit	int	Specifies the number of results to retrieve per page. Acceptable values are 1 to 100, inclusive. (Default: 10)
order	string	Specifies the method to sort a list of results. Acceptable values are the following: - created_at (default): sorts by the time of scheduled message creation in reverse chronological order. The most recently created scheduled message appears first. - scheduled_at: sorts by the time the scheduled message is scheduled to begin in reverse chronological order. The most immediate scheduled message appears first.
reverse	boolean	Determines whether the sort order will be reversed.
status[]	array of objects	Specifies the scheduled message status to retrieve. If not specified, all scheduled messages are returned regardless of their status.
message_type	string	Specifies a message type to retrieve. Acceptable values are MESG, FILE, and ADMM. If not specified, all scheduled messages are returned regardless of their message type.

## Response

If successful, this action returns a scheduled message resource in the response body.

```
{
  "scheduled_messages" : [
    {
      "scheduled_message_id" : 1,
      "message_type" : "MESG",
      "message" : "hi",
      ...
    }
  ],
  "next" : "token for next page"
```

In the case of an error, an error object is returned. A detailed list of error codes is available in our [documentation](#).

---

## Cancel a scheduled message

Cancels a message that a user has scheduled to send at a later time.

### HTTP request

```
DELETE
https://api-{application_id}.sendbird.com/v3/group_channels/{channel_url}/scheduled_messages/{scheduled_message_id}
```

### Parameters

The following table lists the parameters that this action supports.

Required	Type	Description
channel_url	string	Specifies the URL of the channel.
scheduled_message_id	long	Specifies the unique ID of the scheduled message.

### Response

If successful, this action returns a scheduled message resource in the response body.

In the case of an error, an error object is returned. A detailed list of error codes is available in our [documentation](#).

## Update a scheduled message

Updates information on a specific scheduled message. You can't change the message type. Update operation should be done at least 5 minutes prior to the original scheduled time.

## HTTP request

```
PUT
https://api-{application_id}.sendbird.com/v3/group_channels/{channel_url}/scheduled_messages/{scheduled_message_id}
```

## Parameters

The following table lists the parameters that this action supports.

Required	Type	Description
channel_url	string	Specifies the URL of the channel.
scheduled_message_id	long	Specifies the unique ID of the scheduled message.

## Request body

The following table lists the properties of an HTTP request that this action supports for updating a scheduled text message, file message, and admin message.

List of properties for updating a scheduled text message

Optional	Type	Description
message	string	Specifies the content of the message.
scheduled_at	long	The time to send the message, in Unix milliseconds format. Since messages are scheduled in minutes, values less than seconds are discarded. The term between current time and <code>scheduled_at</code> should be between 5 minutes and 30 days.
custom_type	string	Specifies a custom message type used for message grouping. The length is limited to 128 characters.

data	string	Specifies additional message information such as custom font size, font type or JSON formatted string.
send_push	boolean	Determines whether to send a push notification of the message to the channel members. This property only applies to group channels. (Default: true)
mention_type	string	Specifies the mentioning method that indicates which user receives a notification of the message. Acceptable values are users and channels. If set to users, only users specified in the mentioned_user_ids property below are notified. If set to channels, all users in the channel are notified. (Default: users)
mentioned_user_ids[ ]	array of strings	Specifies an array of one or more IDs of the users who will receive a notification of the message.
is_silent	boolean	Determines whether to send a message without updating some of the following channel properties. If set to true, the channel's last_message is updated only for the sender while its unread_message_count remains unchanged for all channel members. Also, a push notification isn't sent to the users receiving the message. If the message is sent to a hidden channel, the channel remains hidden. (Default: false)  * Once the value of this property is set, it's irreversible.
mark_as_read	boolean	Determines whether to mark the message as read for the sender. If set to false, then the sender's unread_count and read_receipt remain unchanged after the message is sent. (Default: true)
sorted_metaarray	array of objects	Specifies a JSON object of one or more key-values items which store additional message information. Each item consists of a key and the values in an array. Items are saved and will be returned in the exact order they've been specified.
dedup_id	string	Specifies the unique message ID created by another system. In general, this property is used to prevent the same message data from getting inserted when migrating messages from another system to Sendbird server. If specified, the server performs a duplicate check using the property value.
apns_bundle_id	string	Specifies the bundle ID of the client app in order to



		send a push notification to iOS devices. You can find this in <b>Settings &gt; Chat &gt; Notifications &gt; Push notification services</b> on the Sendbird Dashboard.
apple_critical_alert_options	object	Specifies options that support <a href="#">Apple's critical alerts</a> and checks whether the message is a critical alert.
sound	string	Specifies the name of a sound file that is used for critical alerts.
volume	float	Specifies the volume of the critical alert sound. The volume ranges from 0.0 to 1.0, which indicates silent and full volume, respectively. (Default: 1.0)

#### List of properties for updating a scheduled file message

Optional	Type	Description
url	string	Specifies the URL of the file which is hosted on the server of your own or other external third party companies. If this url property is specified, the file property is not required.
scheduled_at	long	The time to send the message, in Unix milliseconds format. Since messages are scheduled in minutes, values less than seconds are discarded. The term between current time and scheduled_at should be between 5 minutes and 30 days.
file_name	string	If the file property is used for file upload, this property is automatically specified by Sendbird server. Otherwise, specify the name of the file directly.
file_size	int	If the file property is used for file upload, this property is automatically specified by Sendbird server. Otherwise, specify the size of the file directly.
file_type	string	If the file property is used for file upload, this property is automatically specified by Sendbird server. Otherwise, specify the media type of the file directly.
thumbnails[]	array of strings	Specifies an array of one or more URLs of external thumbnail images that are generated from the image specified by the url property.

custom_type	string	Specifies a custom message type used for message grouping. The length is limited to 128 characters.
data	string	Specifies additional message information such as custom font size, font type or JSON formatted string.
require_auth	boolean	<p>Determines whether the files and thumbnail images in a message are only accessible by users who are in the same channel. This property is only effective when using the share encrypted files feature. This feature encrypts all types of uploaded files and the automatically-generated thumbnail images with AES256, stores them securely in Sendbird server, and enables only the users with access to share the files. You can use this property to restrict access to the files.</p> <p>If set to true, the files are uploaded first, then encrypted, and finally stored along with the thumbnail images which are encrypted as well. When accessing them, the parameter auth should be added to the end of the file URLs. The parameter auth is specified with an encryption key of the user such as ?auth=RW5b2RIHaMdGV4eA==. An encryption key managed internally by the Sendbird server is generated every time a user logs into Sendbird server and is valid for three days starting from the last login. The system mainly uses the key to check if a user is properly accessing the data of the application. (Default: true)</p>
send_push	boolean	Determines whether to send a push notification of the message to the channel members. This property only applies to group channels. (Default: true)
mention_type	string	Specifies the mentioning method that indicates which user receives a notification of the message. Acceptable values are users and channels. If set to users, only users specified in the mentioned_user_ids property below are notified. If set to channels, all users in the channel are notified. (Default: users)
mentioned_user_ids[ ]	array of strings	Specifies an array of one or more IDs of the users who will receive a notification of the message.
is_silent	boolean	Determines whether to send a message without updating some of the following channel properties. If set to true, the channel's last_message is updated only

		<p>for the sender while its <code>unread_message_count</code> remains unchanged for all channel members. Also, a push notification isn't sent to the users receiving the message. If the message is sent to a hidden channel, the channel remains hidden. (Default: false)</p> <p>* Once the value of this property is set, it's irreversible.</p>
<code>mark_as_read</code>	boolean	Determines whether to mark the message as read for the sender. If set to false, then the sender's <code>unread_count</code> and <code>read_receipt</code> remain unchanged after the message is sent. (Default: true)
<code>sorted_metaarray</code>	array of objects	Specifies a <code>JSON</code> object of one or more key-values items which store additional message information. Each item consists of a key and the values in an array. Items are saved and will be returned in the exact order they've been specified.
<code>dedup_id</code>	string	Specifies the unique message ID created by another system. In general, this property is used to prevent the same message data from getting inserted when migrating messages from another system to Sendbird server. If specified, the server performs a duplicate check using the property value.
<code>apns_bundle_id</code>	string	Specifies the bundle ID of the client app in order to send a push notification to iOS devices. You can find this in <code>Settings &gt; Chat &gt; Notifications &gt; Push notification services</code> on the Sendbird Dashboard.
<code>apple_critical_alert_options</code>	object	Specifies options that support <a href="#">Apple's critical alerts</a> and checks whether the message is a critical alert.
<code>sound</code>	string	Specifies the name of a sound file that is used for critical alerts.
<code>volume</code>	float	Specifies the volume of the critical alert sound. The volume ranges from 0.0 to 1.0, which indicates silent and full volume, respectively. (Default: 1.0)

List of properties for updating a scheduled admin message

Optional	Type	Description
----------	------	-------------

message	string	Specifies the content of the message.
scheduled_at	long	The time to send the message, in Unix milliseconds format. Since messages are scheduled in minutes, values less than seconds are discarded. The term between current time and <code>scheduled_at</code> should be between 5 minutes and 30 days.
custom_type	string	Specifies a custom message type used for message grouping. The length is limited to 128 characters.
data	string	Specifies additional message information such as custom font size, font type or JSON formatted string.
send_push	boolean	Determines whether to send a push notification of the message to the channel members. This property only applies to group channels. (Default: true)
mention_type	string	Specifies the mentioning method that indicates which user receives a notification of the message. Acceptable values are <code>users</code> and <code>channels</code> . If set to <code>users</code> , only users specified in the <code>mentioned_user_ids</code> property below are notified. If set to <code>channels</code> , all users in the channel are notified. (Default: <code>users</code> )
mentioned_user_ids[ ]	array of strings	Specifies an array of one or more IDs of the users who will receive a notification of the message.
is_silent	boolean	Determines whether to send a message without updating some of the following channel properties. If set to true, the channel's <code>last_message</code> is updated only for the sender while its <code>unread_message_count</code> remains unchanged for all channel members. Also, a push notification isn't sent to the users receiving the message. If the message is sent to a hidden channel, the channel remains hidden. (Default: false)  * Once the value of this property is set, it's irreversible.
sorted_metaarray	array of objects	Specifies a JSON object of one or more key-values items which store additional message information. Each item consists of a key and the values in an array. Items are saved and will be returned in the exact order they've been specified.

dedup_id	string	Specifies the unique message ID created by another system. In general, this property is used to prevent the same message data from getting inserted when migrating messages from another system to Sendbird server. If specified, the server performs a duplicate check using the property value.
apns_bundle_id	string	Specifies the bundle ID of the client app in order to send a push notification to iOS devices. You can find this in Settings > Chat > Notifications > Push notification services on the Sendbird Dashboard.

## Response

If successful, this action returns a list of scheduled messages resource in the response body.

In the case of an error, an error object is returned. A detailed list of error codes is available in our [documentation](#).

## Send a scheduled message immediately

Send a scheduled message immediately. A user can only send their own scheduled messages immediately.

### HTTP request

```
POST
https://api-{application_id}.sendbird.com/v3/{channel_type}/{channel_url}/scheduled_messages/{scheduled_message_id}/send_now
```

### Parameters

The following table lists the parameters that this action supports.

Optional	Type	Description
----------	------	-------------

channel_type	string	Specifies the channel type of the channel.
channel_url	string	Specifies the URL of the channel.
scheduled_message_id	long	Specifies the unique ID of the scheduled message.

## Response

If successful, this action returns an empty body.

In the case of an error, an error object is returned. A detailed list of error codes is available in our [documentation](#).

## Retrieve the total number of schedule messages

Retrieves the total number of scheduled messages that a user has.

## HTTP request

```
GET
https://api-{application_id}.sendbird.com/v3/scheduled_messages/count
```

## Parameters

The following table lists the parameters that this action supports.

Required	Type	Description
channel_type	string	Specifies the channel type of the channel.

channel_url	string	Specifies the URL of the channel.
sender_id	string	Specifies the user ID of the sender.
status[]	array of objects	Specifies the scheduled message status to retrieve. If not specified, all scheduled messages are returned regardless of their status.
message_type	string	Specifies a message type to retrieve. Acceptable values are <code>MESG</code> , <code>FILE</code> , and <code>ADMM</code> . If not specified, all scheduled messages are returned regardless of their message type.

## Response

If successful, this action returns a count of scheduled messages

```
{
  "count": 50
}
"next" : "token for next page"
```

In the case of an error, an error object is returned. A detailed list of error codes is available in our [documentation](#).

## SDK specifications

### Android:

#### Create a scheduled message

You can create a scheduled user message to send at a later time by passing `ScheduledUserMessageCreateParams` as an argument to the `groupChannel.createScheduledUserMessage()` method.

```
// create scheduled user message
groupChannel.createScheduledUserMessage(
    scheduledUserMessageCreateParams: ScheduledUserMessageCreateParams,
    handler: UserMessageHandler?
)
```

### ScheduledUserMessageCreateParams

```
class ScheduledUserMessageCreateParams {
    scheduledAt: Long
    message: String = ""
    translationTargetLanguages: List<String>? = null
    data: String? = null
    customType: String? = null
    mentionType: MentionType = MentionType.USERS
    mentionedUserIds: List<String>?
    metaArrays: List<MessageMetaArray>?
    appleCriticalAlertOptions: AppleCriticalAlertOptions? = null
    pushNotificationDeliveryOption: PushNotificationDeliveryOption? = null
}
```

You can create a scheduled file message to send at a later time by passing `ScheduledFileMessageCreateParams` as an argument to the `groupChannel.createScheduledFileMessage()` method.

```
// create scheduled file message
groupChannel.createScheduledFileMessage(
    scheduledFileMessageCreateParams: ScheduledFileMessageCreateParams,
    handler: FileMessageHandler?
)
```

### ScheduledFileMessageCreateParams

```
class ScheduledFileMessageUpdateParams {
    scheduledAt: Long
    fileUrl: String?
```



```
file: File?
fileName: String? = null
mimeType: String? = null
fileSize: Int? = null
thumbnailSizes: List<FileMessage.ThumbnailSize>? = null
data: String? = null
customType: String? = null
mentionType: MentionType = MentionType.USERS
mentionedUserIds: List<String>? = null
metaArrays: List<MessageMetaArray>? = null
appleCriticalAlertOptions: AppleCriticalAlertOptions? = null
pushNotificationDeliveryOption: PushNotificationDeliveryOption? = null
}
```

---

## Update a scheduled message

You can update a scheduled user message by passing `ScheduledUserMessageUpdateParams` as an argument to the `groupChannel.updateScheduledUserMessage()` method.

```
// update scheduled user message

groupChannel.updateScheduledUserMessage(
    scheduledMessageId: Long,
    scheduledUserMessageUpdateParams: ScheduledUserMessageUpdateParams,
    handler: UserMessageHandler?
)
```

### ScheduledUserMessageUpdateParams

```
class ScheduledUserMessageUpdateParams {
    scheduledAt: Long? = null
    message: String? = null
    translationTargetLanguages: List<String>? = null
    data: String? = null
    customType: String? = null
    mentionType: MentionType = MentionType.USERS
    mentionedUserIds: List<String>? = null
    metaArrays: List<MessageMetaArray>? = null
}
```

```
appleCriticalAlertOptions: AppleCriticalAlertOptions? = null
pushNotificationDeliveryOption: PushNotificationDeliveryOption? = null
}
```

You can update a scheduled file message by passing `ScheduledFileMessageUpdateParams` as an argument to the `groupChannel.updateScheduledFileMessage()` method.

```
// update scheduled file message
groupChannel.updateScheduledFileMessage(
    scheduledMessageId: Long,
    scheduledFileMessageUpdateParams: ScheduledFileMessageUpdateParams,
    handler: FileMessageHandler?
)
```

`ScheduledFileMessageUpdateParams`

```
class ScheduledFileMessageUpdateParams {
    scheduledAt: Long? = null
    fileUrl: String? = null
    file: File? = null
    fileName: String? = null
    mimeType: String? = null
    fileSize: Int? = null
    thumbnailSizes: List<FileMessage.ThumbnailSize>? = null
    data: String? = null
    customType: String? = null
    mentionType: MentionType = MentionType.USERS
    mentionedUserIds: List<String>? = null
    metaArrays: List<MessageMetaArray>? = null
    appleCriticalAlertOptions: AppleCriticalAlertOptions? = null
    pushNotificationDeliveryOption: PushNotificationDeliveryOption? = null
}
```

---

## Cancel a scheduled message

---

Using the `groupChannel.cancelScheduledMessage()` method, you can cancel a message that a user has scheduled to send at a later time.

```
// cancel a scheduled message

groupChannel.cancelScheduledMessage(
  scheduledMessageId: Long,
  handler: CompletionHandler?
)
```

---

## Send a scheduled message immediately

Using the `groupChannel.sendScheduledMessageNow()` method, you can send a scheduled message immediately instead of the message being sent at the scheduled time.

```
// send a scheduled message immediately

groupChannel.sendScheduledMessageNow(
  scheduledMessageId: Long,
  handler: CompletionHandler?
)
```

---

## Retrieve a scheduled message

You can retrieve a specific scheduled message by passing `ScheduledMessageRetrievalParams` as an argument to the `BaseMessage.getScheduledMessage()` method.

```
// Retrieve a scheduled message

BaseMessage.getScheduledMessage(
  params: ScheduledMessageRetrievalParams,
  handler: BaseMessageHandler?
)
```

---

## ScheduledMessageRetrievalParams

```
class ScheduledMessageRetrievalParams {
    channelUrl: String
    scheduledMessageId: Long
}
```

---

## List all scheduled messages

You can retrieve all scheduled messages by using the `ScheduledMessageListQuery`.

```
class ScheduledMessageListQuery {
    limit: Int = 20
    channelUrl: String? = null
    order: Order = Order.CREATED_AT // Order.SCHEDULED_AT
    reverse: Boolean = false
    scheduledStatus: List<ScheduledStatus>? = null
    messageTypeFilter: MessageTypeFilter = MessageTypeFilter.ALL // USER, FILE,
ADMIN, ALL

    fun next(handler: BaseMessagesHandler?)
}
```

## ScheduledMessageListQuery

Property name	Type	Description
limit	int	Specifies the number of results to return per call. Acceptable values are 1 to 100, inclusive. The recommended value for this parameter is 30. (Default: 20)
reverse	boolean	Determines whether to sort the retrieved messages in reverse order. If <code>false</code> , the results are in ascending order.
messageTypeFilter	enum	Specifies the message type to filter the messages with the corresponding type. Acceptable values are

---

		MessageTypeFilter.ALL, MessageTypeFilter.User, MessageTypeFilter.FILE, and MessageTypeFilter.ADMIN.
--	--	---

---

## Retrieve the total number of scheduled messages

Using the `SendbirdChat.getTotalScheduledMessageCount()` method, you can retrieve the total number of scheduled messages a user has. The properties of `TotalScheduledMessageCountParams` are optional to set as a filter. If not set, the number of all scheduled messages that a user has will be retrieved.

```
// Retrieve the total number of scheduled message

SendbirdChat.getTotalScheduledMessageCount(
    params: TotalScheduledMessageCountParams,
    handler: CountHandler?
)
```

TotalScheduledMessageCountParams

```
class TotalScheduledMessageCountParams {
    channelUrl: String? = null
    scheduledStatus: List<ScheduledStatus>? = null
    messageTypeFilter: MessageTypeFilter = MessageTypeFilter.ALL // USER, FILE,
ADMIN, ALL
}
```

## iOS (Swift):

### Create a scheduled message

You can create a scheduled user message to send at a later time by passing `ScheduledUserMessageCreateParams` as an argument to the `groupChannel.createScheduledUserMessage()` method.

```
// Creates a scheduled user message.
```

```
let params = ScheduledUserMessageCreateParams(  
    scheduledAt: SCHEDULED_AT_MILLISECONDS,  
    message: MESSAGE_TEXT  
)  
groupChannel.createScheduledUserMessage(params: params) { message, error in  
    guard error == nil else {  
        return // Handle error.  
    }  
}
```

#### ScheduledBaseMessageCreateParams

```
public class ScheduledBaseMessageCreateParams: NSObject, Codable {  
  
    /// The time to send the message, in Unix milliseconds format. The Time between  
    /// current time and `scheduledAt` should be more than 5 minutes.  
    public var scheduledAt: Int64  
  
    /// The message data. The default value is nil.  
    public var data: String?  
  
    /// The custom type of the message. The default value is `nil`.  
    public var customType: String?  
  
    /// The push notification delivery option that determines how to deliver the push  
    /// notification when sending
```

```
/// a user or a file message. The default value is `.default`.

public var pushNotificationDeliveryOption: PushNotificationDeliveryOption =
    .default

/// The meta arrays with keys and values

public var metaArrays: [MessageMetaArray]?

/// The mention type that represents the target type of mention.

public var mentionType: MentionType = .users

/// The mentioned user IDs.

public var mentionedUserIds: [String]?

/// Mention to specific users with user objects.

public var mentionedUsers: [User]?

/// Adds user IDs for mention

public func addMentionedUserIds(_ userIds: [String])

/// The push notification options for the Apple critical alert. The default value
is `nil`.

public var appleCriticalAlertOptions: AppleCriticalAlertOptions?

public init(scheduledAt: Int64)

}
```

#### ScheduledUserMessageCreateParams

```
public class ScheduledUserMessageCreateParams: ScheduledBaseMessageCreateParams {
```

```
/// The message text.  
  
public var message: String  
  
/// The target languages that the message will be translated into.  
  
public var translationTargetLanguages: [String]?  
  
/// Initializes and returns a params object for creating a scheduled user  
message.  
  
public init(scheduledAt: Int64, message: String)  
  
}
```

You can create a scheduled file message to send at a later time by passing `ScheduledFileMessageCreateParams` as an argument to the `groupChannel.createScheduledFileMessage()` method.

```
// Creates a scheduled file message.  
  
let params = ScheduledFileMessageCreateParams(  
  
    scheduledAt: SCHEDULED_AT_MILLISECONDS,  
    file: FILE_DATA  
  
)  
  
params.fileName = FILE_NAME  
params.fileSize = FILE_SIZE  
params.mimeType = MIME_TYPE  
  
groupChannel.createScheduledFileMessage(params: params) { message, error in  
    guard error == nil else {  
        return // Handle error.  
    }  
  
}
```



---

## ScheduledFileMessageCreateParams

```
public class ScheduledFileMessageCreateParams: ScheduledBaseMessageCreateParams {  
  
    /// The binary file data. `file` and `fileURL` cannot be set together.  
  
    public var file: Data?  
  
    /// The file URL. `file` and `fileURL` cannot be set together.  
  
    public var fileURL: String?  
  
    /// The thumbnail sizes. This parameter is the array of `ThumbnailSize`  
instance and works for image file only.  
  
    public var thumbnailSizes: [ThumbnailSize]?  
  
    /// The file name.  
  
    public var fileName: String?  
  
    /// The file MIME type.  
  
    public var mimeType: String?  
  
    /// The file size.  
  
    public var fileSize: UInt = 0  
  
    /// Initializes an instance of a file message params with binary file.  
  
    public init(scheduledAt: Int64, file: Data)  
  
    /// Initializes and returns a params object for creating a scheduled file  
message with the given file URL.  
  
    public init(scheduledAt: Int64, fileURL: String)  
  
}
```

---

## Update a scheduled message

You can update a scheduled user message by passing `ScheduledUserMessageUpdateParams` as an argument to the `groupChannel.updateScheduledUserMessage()` method.

```
let params = ScheduledUserMessageUpdateParams()
params.scheduledAt = SCHEDULED_AT_MILLISECONDS
params.translationTargetLanguages = TRANSLATION_TARGET_LANGUAGES
params.data = CUSTOM_DATA
params.customType = CUSTOM_TYPE

groupChannel.updateScheduledUserMessage(scheduledMessageId:
SCHEDULED_MESSAGE_ID, params: params) { message, error in
    guard error == nil else {
        return // Handle error.
    }
}
```

#### ScheduledBaseMessageUpdateParams

```
public class ScheduledBaseMessageUpdateParams: NSObject {

    /// The time to send the message, in Unix milliseconds format. The time between
    current time and `scheduledAt` should be more than 5 minutes.

    public var scheduledAt: Int64 = 0

    /// The message data. The default value is nil.

    public var data: String?

    /// The custom type of the message. The default value is `nil`.

    public var customType: String?

    /// The push notification delivery option that determines how to deliver the
    push notification when sending

    /// a user or a file message. The default value is `.default`.

    public var pushNotificationDeliveryOption: PushNotificationDeliveryOption =
.default
```

```
/// The meta arrays with keys and values.
public var metaArrays: [MessageMetaArray]?

/// The mention type that represents target type of mention.
public var mentionType: MentionType = .users

/// The mentioned user IDs.
public var mentionedUserIds: [String]?

/// Sets the `mentionedUserIds` by this method either.
public func setMentionedUsers(_ mentionedUsers: [User])

/// Adds user IDs for mention
public func addMentionedUserIds(_ userIds: [String])

/// The push notification options for the Apple critical alert. The default
value is `nil`.
public var appleCriticalAlertOptions: AppleCriticalAlertOptions?
}
```

#### ScheduledUserMessageUpdateParams

```
public class ScheduledUserMessageUpdateParams: ScheduledBaseMessageUpdateParams {

/// The message text.
public var message: String?

/// The target languages that the message will be translated into.
public var translationTargetLanguages: [String]?

/// The custom payload.
public var payload: [String: Any] = [:]

/// Initializes and returns a params object for updating a scheduled user
message.
public override init()

}
```

You can update a scheduled file message by passing `ScheduledFileMessageUpdateParams` as an argument to the `groupChannel.updateScheduledFileMessage()` method.

```
// Updates scheduled file message.

let params = ScheduledFileMessageUpdateParams()
params.scheduledAt = SCHEDULED_AT_MILLISECONDS
params.file = FILE_DATA
params.fileName = FILE_NAME
params.fileSize = FILE_SIZE
params.mimeType = MIME_TYPE

groupChannel.updateScheduledFileMessage(scheduledMessageId: SCHEDULED_MESSAGE_ID,
params: params) { message, error in
    guard error == nil else {
        return // Handle error.
    }
}

}
```

### ScheduledFileMessageUpdateParams

```
public class ScheduledFileMessageUpdateParams: ScheduledBaseMessageUpdateParams {

    /// The binary file data. `file` and `fileURL` cannot be set together.
    public var file: Data?

    /// The file URL. `file` and `fileURL` cannot be set together.
    public var fileURL: String?

    /// The thumbnail sizes. This parameter is the array of `ThumbnailSize`
    instance and works for image file only.
    public var thumbnailSizes: [ThumbnailSize]?

    /// The file name.
    public var fileName: String?

    /// The file MIME type.
    public var mimeType: String?

}
```

```
/// The file size.
public var fileSize: UInt = 0

/// Initializes and returns a params object for updating a scheduled file
message.
public override init()

}
```

---

## Cancel a scheduled message

Using the `groupChannel.cancelScheduledMessage()` method, you can cancel a message that a user has scheduled to send at a later time.

```
// Cancels a scheduled message.

groupChannel.cancelScheduledMessage(scheduledMessageId: SCHEDULED_MESSAGE_ID) {
message, error in
    guard error == nil else {
        return // Handle error.
    }
}
```

---

## Send a scheduled message immediately

Using the `groupChannel.sendScheduledMessageNow()` method, you can send a scheduled message immediately instead of the message being sent at the scheduled time.

```
groupChannel.sendScheduledMessageNow(scheduledMessageId: SCHEDULED_MESSAGE_ID) {
```

```
error in

    guard error == nil else {

        return // Handle error.

    }

}
```

---

## Retrieve a scheduled message

You can retrieve a specific scheduled message by passing `ScheduledMessageRetrievalParams` as an argument to the `BaseMessage.getScheduledMessage()` method.

```
// Retrieves a scheduled message.

let params = ScheduledMessageRetrievalParams(

    channelURL: CHANNEL_URL,
    scheduledMessageId: SCHEDULED_MESSAGE_ID
)

BaseMessage.getScheduledMessage(params: params) { message, error in
    guard error == nil else {
        return // Handle error.
    }
}
```

ScheduledMessageRetrievalParams

```
public class ScheduledMessageRetrievalParams: NSObject {

    /// Specifies the URL of the channel to retrieve the message.
    public let channelURL: String
```

```
/// Specifies the unique ID of the scheduled message to retrieve.
public let scheduledMessageId: Int64

/// Initializes and returns a params object for retrieving scheduled message.
public init(channelURL: String, scheduledMessageId: Int64)
}
```

## List all scheduled messages

You can retrieve all scheduled messages by using the `ScheduledMessageListQuery`.

```
// If `CHANNEL_URL` is nil, you will get scheduled messages for all channels.

let query = SendbirdChat.createScheduledMessageListQuery(channelURL: CHANNEL_URL) {
  params in
    params.limit = LIMIT_PER_PAGE
    params.order = SCHEDULED_MESSAGE_LIST_ORDER
    params.reverse = REVERSE_RESULT_LIST
    params.scheduledStatusOptions = SCHEDULED_STATUS_OPTIONS
    params.messageTypeFilter = MESSAGE_TYPE_FILTER
}

query.loadNextPage { messages, error in
  guard error == nil else {
    return // Handle error.
  }
}
```

```
public class ScheduledMessageListQuery: NSObject {

  /// A Boolean value that indicates whether there are more data to fetch.
  public private(set) var hasNext: Bool = true

  /// A Boolean value that indicates the query is currently loading or not.
  public private(set) var isLoading: Bool = false
```

```

/// Fetches next page.
public func loadNextPage(completionHandler: @escaping MessageListHandler)

}

```

```

public final class ScheduledMessageListQueryParams: NSObject {

    /// The maximum number of `BaseMessage`s per queried page. Default is `20`.
    public var limit: Int = 20

    /// The order method for the search. Default is `.createdAt`
    public var order: ScheduledMessageListOrder = .createdAt

    /// A Boolean value that indicates whether to reverse the queried result list.
    Default is `false`.
    public var reverse: Bool = false

    /// Restricts the search scope only to retrieve the messages with the specified
    message status. If not specified, all messages are retrieved.
    public var scheduledStatusOptions: ScheduledStatusOptions = []

    /// Restricts the search scope only to retrieve messages with the specified
    message type. Default is `.all`.
    public var messageTypeFilter: MessageTypeFilter = .all

    /// Default constructor.
    public override init()

    /// Initializes and returns a newly allocated params object that mutated
    through builder closure.
    public init(builder: (ScheduledMessageListQueryParams) -> Void)

}

```

### ScheduledMessageListQuery

Property name	Type	Description
limit	int	Specifies the number of results to return per call. Acceptable values are 1 to 100, inclusive. The recommended value for this parameter is 30. (Default: 20)



---

reverse	boolean	Determines whether to sort the retrieved messages in reverse order. If false, the results are in ascending order.
messageTypeFilter	enum	Specifies the message type to filter the messages with the corresponding type. Acceptable values are <code>MessageTypeFilter.ALL</code> , <code>MessageTypeFilter.User</code> , <code>MessageTypeFilter.FILE</code> , and <code>MessageTypeFilter.ADMIN</code> .

---

## Retrieve the total number of scheduled messages

Using the `SendbirdChat.getTotalScheduledMessageCount()` method, you can retrieve the total number of scheduled messages a user has. The properties of `TotalScheduledMessageCountParams` are optional to set as a filter. If not set, the number of all scheduled messages that a user has will be retrieved.

```
// Retrieves the total number of scheduled messages.

let params = TotalScheduledMessageCountParams(channelURL: CHANNEL_URL)
params.messageTypeFilter = MESSAGE_TYPE_FILTER

params.scheduledStatusOptions = SCHEDULED_STATUS_OPTIONS

SendbirdChat.getTotalScheduledMessageCount(params: params) { count, error in
    guard error == nil else {
        return // Handle error.
    }
}
```

TotalScheduledMessageCountParams

```
public class TotalScheduledMessageCountParams: NSObject {

    /// Channel URL which has this message.
    public let channelURL: String

    /// Restricts the search scope only to retrieve the messages with the specified
    message status. If not specified, all messages are retrieved.
    public var scheduledStatusOptions: ScheduledStatusOptions = []
}
```

```
    /// Restricts the search scope only to retrieve messages with the specified
    message type. Default is `.all`.
    public var messageTypeFilter: MessageTypeFilter = .all

    /// Initializes and returns a params object for getting total scheduled message
    count.
    public init(channelURL: String)
  }
```

## Javascript (Typescript):

### Create a scheduled message

You can create a scheduled user message to send at a later time by passing `ScheduledUserMessageCreateParams` as an argument to the `groupChannel.createScheduledUserMessage()` method.

```
// Creates a scheduled user message.
const params: ScheduledUserMessageCreateParams = {
```

```
    message: 'Off work, WOOHOO!',
    scheduledAt: TODAY_AT_SIX_PM,
  });
channel.createScheduledUserMessage(params)
  .onPending((message: SendableMessage) => {
    // Pending scheduling message is received.
  })

  .onSucceeded((message: SendableMessage) => {
    // Scheduled message is received.
  })

  .onFailed((err: Error, message: SendableMessage) => {
    // error is received. Failed message is null.
  });

const handler: GroupChannelHandler = new GroupChannelHandler();
handler.onMessageReceived = (channel: BaseChannel, message: BaseMessage) => {
  // The sent scheduled message is received.
};
sb.groupChannel.addGroupChannelHandler('groupChannelHandler', handler);
```

#### ScheduledUserMessageCreateParams

```
interface ScheduledUserMessageCreateParams extends UserMessageCreateParams {
  scheduledAt: number;
}
```

You can create a scheduled file message to send at a later time by passing `ScheduledFileMessageCreateParams` as an argument to the `groupChannel.createScheduledFileMessage()` method.

```
// Creates a scheduled file message.

const params: ScheduledFileMessageCreateParams = {
  file: FILE,
  scheduledAt: TODAY_AT_SIX_PM,
};
channel.createScheduledFileMessage(params)
  .onPending((message: SendableMessage) => {
    // Pending scheduling message is received.
  })
```

```
.onSucceeded((message: SendableMessage) => {
  // Scheduled message is received.
})
.onFailed((err: Error, message: SendableMessage) => {
  // error is received. Failed message is null.
});

const handler: GroupChannelHandler = new GroupChannelHandler();
handler.onMessageReceived = (channel: BaseChannel, message: BaseMessage) => {
  // The sent scheduled message is received.
};
sb.groupChannel.addGroupChannelHandler('groupChannelHandler', handler);
```

ScheduledFileMessageCreateParams

```
interface ScheduledFileMessageCreateParams extends FileMessageCreateParams {
  scheduledAt: number;
}
```

---

## Update a scheduled message

You can update a scheduled user message by passing `ScheduledUserMessageUpdateParams` as an argument to the `groupChannel.updateScheduledUserMessage()` method.

```
const params: ScheduledUserMessageUpdateParams = {
  message: 'Late shift, sad...',
  scheduledAt: TODAY_AT_SEVEN_PM,
};

// Updates a scheduled user message.
const updatedScheduledMessage: UserMessage = await
channel.updateScheduledUserMessage(
  scheduledMessage.scheduledInfo.scheduledMessageId,
  params,
);
```

ScheduledUserMessageUpdateParams

```
interface ScheduledUserMessageUpdateParams extends UserMessageUpdateParams {
  scheduledAt?: number; // null
}
```

You can update a scheduled file message by passing `ScheduledFileMessageUpdateParams` as an argument to the `groupChannel.updateScheduledFileMessage()` method.

```
const params: ScheduledFileMessageUpdateParams = {
  scheduledAt: TODAY_AT_SEVEN_PM,
};

// Updates a scheduled file message.

const updatedScheduledMessage: FileMessage = await
channel.updateScheduledFileMessage(
  scheduledMessage.scheduledInfo.scheduledMessageId,
  params,
);
```

#### ScheduledFileMessageUpdateParams

```
interface ScheduledFileMessageUpdateParams extends BaseMessageUpdateParams {
  scheduledAt?: number;
  file?: FileCompat;
  fileUrl?: string;
  fileName?: string;
  fileSize?: number;
  mimeType?: string;
  thumbnailSizes?: ThumbnailSize[];
  requireAuth?: boolean;
}
```

---

## Cancel a scheduled message

---

Using the `groupChannel.cancelScheduledMessage()` method, you can cancel a message that a user has scheduled to send at a later time.

```
// Cancels a scheduled message.  
await groupChannel.cancelScheduledMessage(scheduledMessageId);
```

---

## Send a scheduled message immediately

Using the `groupChannel.sendScheduledMessageNow()` method, you can send a scheduled message immediately instead of the message being sent at the scheduled time.

```
// Send a scheduled message immediately.  
await  
channel.sendScheduledMessageNow(scheduledMessage.scheduledInfo.scheduledMessageId);
```

---

## Retrieve a scheduled message

You can retrieve a specific scheduled message by passing `ScheduledMessageRetrievalParams` as an argument to the `MessageModule.getScheduledMessage()` method.

```
const params: ScheduledMessageRetrievalParams = {  
  channelUrl: channel.url,  
  scheduledMessageId: scheduledMessage.scheduledInfo.scheduledMessageId,  
};  
const scheduledMessage: BaseMessage = await sb.message.getScheduledMessage(params);
```

ScheduledMessageRetrievalParams

```
interface ScheduledMessageRetrievalParams {  
  channelUrl: string; // Specifies the URL of a channel.  
  scheduledMessageId: number; // Specifies the ID of the scheduled message.  
}
```

## List all scheduled messages

You can retrieve all scheduled messages by using the `ScheduledMessageListQuery`.

```
const params: ScheduledMessageListQueryParams = {
  channelId: CHANNEL_URL,
  scheduledStatus: [ScheduledStatus.PENDING],
};

const scheduledMessageListQuery: ScheduledMessageListQuery =
  sb.groupChannel.createScheduledMessageListQuery(params);

const queriedScheduledMessages: BaseMessage[] = await
  scheduledMessageListQuery.next();
```

### ScheduledMessageListQueryParams

```
interface ScheduledMessageListQueryParams extends BaseListQueryParams {
  channelId?: string; // null
  order?: ScheduledMessageListOrder; // null
  reverse?: boolean; // false
  scheduledStatus?: ScheduledStatus[]; // null
  messageTypeFilter?: MessageTypeFilter; // MessageTypeFilter.ALL;
}
```

### ScheduledMessageListQuery

Property name	Type	Description
limit	int	Specifies the number of results to return per call. Acceptable values are 1 to 100, inclusive. The recommended value for this parameter is 30. (Default: 20)
reverse	boolean	Determines whether to sort the retrieved messages in reverse order. If false, the results are in ascending order.
messageTypeFilter	enum	Specifies the message type to filter the messages with the corresponding type. Acceptable values are <code>MessageTypeFilter.ALL</code> , <code>MessageTypeFilter.User</code> , <code>MessageTypeFilter.FILE</code> , and <code>MessageTypeFilter.ADMIN</code> .

---

---

## Retrieve the total number of scheduled messages

Using the `SendbirdChat.getTotalScheduledMessageCount()` method, you can retrieve the total number of scheduled messages a user has. The properties of `TotalScheduledMessageCountParams` are optional to set as a filter. If not set, the number of all scheduled messages that a user has will be retrieved.

```
const params: TotalScheduledMessageCountParams = {
  scheduledStatus: [ScheduledStatus.PENDING],
};

const totalScheduledMessageCount: number = await
sb.groupChannel.getTotalScheduledMessageCount(params);
```

TotalScheduledMessageCountParams

```
interface TotalScheduledMessageCountParams {
  channelUrl?: string; // null
  scheduledStatus?: ScheduledStatus[]; // null
  messageTypeFilter?: MessageTypeFilter; // MessageTypeFilter.ALL
}
```