

[whatsapp-web.js 1.23.0](#)

whatsapp-web.js 1.23.0

npm v1.23.0 dependencies stale WhatsApp Web 2.2346.52 chat 1130 online

whatsapp-web.js

A WhatsApp API client that connects through the WhatsApp Web browser app

It uses Puppeteer to run a real instance of Whatsapp Web to avoid getting blocked.

NOTE: I can't guarantee you will not be blocked by using this method, although it has worked for me. WhatsApp does not allow bots or unofficial clients on their platform, so this shouldn't be considered totally safe.

Quick Links

- [Guide / Getting Started](#) (*work in progress*)
- [Reference documentation](#)
- [GitHub](#)
- [npm](#)

Installation

The module is now available on npm! `npm i whatsapp-web.js`

Please note that Node v12+ is required.

Example usage

```
const { Client } = require('whatsapp-web.js');

const client = new Client();

client.on('qr', (qr) => {
  // Generate and scan this code with your phone
  console.log('QR RECEIVED', qr);
});

client.on('ready', () => {
  console.log('Client is ready!');
});

client.on('message', msg => {
  if (msg.body == '!ping') {
    msg.reply('pong');
  }
});
```

```
}  
});
```

```
client.initialize();
```

Take a look at [example.js](#) for another example with more use cases.

For more information on saving and restoring sessions, check out the available [Authentication Strategies](#).

Supported features

Feature	Status
Multi Device	
Send messages	
Receive messages	
Send media (images/audio/documents)	
Send media (video)	(requires google chrome)
Send stickers	
Receive media (images/audio/video/documents)	
Send contact cards	
Send location	
Send buttons	
Send lists	(business accounts not supported)
Receive location	
Message replies	
Join groups by invite	
Get invite for group	
Modify group info (subject, description)	
Modify group settings (send messages, edit info)	
Add group participants	
Kick group participants	
Promote/demote group participants	
Mention users	
Mute/unmute chats	
Block/unblock contacts	
Get contact info	
Get profile pictures	
Set user status message	
React to messages	

Something missing? Make an issue and let us know!

Contributing

Pull requests are welcome! If you see something you'd like to add, please do. For drastic changes, please open an issue first.

Supporting the project

You can support the maintainer of this project through the links below

- [Support via GitHub Sponsors](#)
- [Support via PayPal](#)
- [Sign up for DigitalOcean](#) and get \$200 in credit when you sign up (Referral)

Disclaimer

This project is not affiliated, associated, authorized, endorsed by, or in any way officially connected with WhatsApp or any of its subsidiaries or its affiliates. The official WhatsApp website can be found at <https://whatsapp.com>. "WhatsApp" as well as related names, marks, emblems and images are registered trademarks of their respective owners.

License

Copyright 2019 Pedro S Lopez

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this project except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>.

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Base

[Base\(\)](#)

BaseAuthStrategy

[BaseAuthStrategy\(\)](#)

BusinessContact

[BusinessContact\(\)](#)
[BusinessContact# block\(\)](#)
[BusinessContact# businessProfile](#)
[BusinessContact# getAbout\(\)](#)
[BusinessContact# getChat\(\)](#)
[BusinessContact# getCommonGroups\(\)](#)
[BusinessContact# getCountryCode\(\)](#)
[BusinessContact# getFormattedNumber\(\)](#)

[BusinessContact# getProfilePicUrl\(\)](#)
[BusinessContact# id](#)
[BusinessContact# isBlocked](#)
[BusinessContact# isBusiness](#)
[BusinessContact# isEnterprise](#)
[BusinessContact# isGroup](#)
[BusinessContact# isMe](#)
[BusinessContact# isMyContact](#)

[BusinessContact# isUser](#)
[BusinessContact# isWAContact](#)
[BusinessContact# name](#)
[BusinessContact# number](#)
[BusinessContact# pushname](#)
[BusinessContact# shortName](#)
[BusinessContact# unblock\(\)](#)

Buttons

[Buttons\(body, buttons, title, footer\)](#)
[Buttons# _format\(buttons\)](#)

[Buttons# body](#)
[Buttons# buttons](#)

[Buttons# footer](#)
[Buttons# title](#)

Call

[Call\(\)](#)
[Call# canHandleLocally](#)
[Call# from](#)
[Call# fromMe](#)

[Call# id](#)
[Call# isGroup](#)
[Call# isVideo](#)
[Call# participants](#)

[Call# reject\(\)](#)

[Call# timestamp](#)
[Call# webClientShouldHandle](#)

Chat

[Chat\(\)](#)
[Chat# archive\(\)](#)
[Chat# archived](#)
[Chat# changeLabels\(labelIds\)](#)
[Chat# clearMessages\(\)](#)
[Chat# clearState\(\)](#)
[Chat# delete\(\)](#)
[Chat# fetchMessages\(searchOptions\)](#)
[Chat# getContact\(\)](#)
[Chat# getLabels\(\)](#)

[Chat# id](#)
[Chat# isGroup](#)
[Chat# isMuted](#)
[Chat# isReadOnly](#)
[Chat# lastMessage](#)
[Chat# markUnread\(\)](#)
[Chat# mute\(unmuteDate\)](#)
[Chat# muteExpiration](#)
[Chat# name](#)
[Chat# pin\(\)](#)

[Chat# pinned](#)
[Chat# sendMessage\(content\[, options\]\)](#)
[Chat# sendSeen\(\)](#)
[Chat# sendStateRecording\(\)](#)
[Chat# sendStateTyping\(\)](#)
[Chat# timestamp](#)
[Chat# unarchive\(\)](#)
[Chat# unmute\(\)](#)
[Chat# unpin\(\)](#)
[Chat# unreadCount](#)

ChatTypes

[ChatTypes. GROUP](#)

[ChatTypes. SOLO](#)

[ChatTypes. UNKNOWN](#)

Client

[Client\(options\)](#)
[Client# acceptGroupV4Invite\(inviteInfo\)](#)
[Client# acceptInvite\(inviteCode\)](#)
[Client# addOrRemoveLabels\(labelIds, chatIds\)](#)

Client# approveGroupMembershipRequests(groupId, options)
Client# archiveChat()
Client# createGroup(title, participants, options)
Client# deleteProfilePicture()
Client# destroy()
Client# event:auth_failure
Client# event:authenticated
Client# event:change_battery
Client# event:change_state
Client# event:chat_archived
Client# event:chat_removed
Client# event:contact_changed
Client# event:disconnected
Client# event:group_admin_changed
Client# event:group_join
Client# event:group_leave
Client# event:group_membership_request
Client# event:group_update
Client# event:incoming_call
Client# event:media_uploaded

Client# event:message
Client# event:message_ack
Client# event:message_create
Client# event:message_edit
Client# event:message_reaction
Client# event:message_revoke_everyone
Client# event:message_revoke_me
Client# event:qr
Client# event:ready
Client# getBlockedContacts()
Client# getChatById(chatId)
Client# getChatLabels(chatId)
Client# getChats()
Client# getChatsByLabelId(labelId)
Client# getCommonGroups(contactId)
Client# getContactById(contactId)
Client# getContacts()
Client# getCountryCode(number)
Client# getFormattedNumber(number)
Client# getGroupMembershipRequests(groupId)
Client# getInviteInfo(inviteCode)
Client# getLabelById(labelId)
Client# getLabels()
Client# getNumberId(number)

Client# getProfilePicUrl(contactId)
Client# getState()
Client# getWebVersion()
Client# info
Client# initialize()
Client# isRegisteredUser(id)
Client# logout()

[Client# markChatUnread\(chatId\)](#)
[Client# muteChat\(chatId, unmuteDate\)](#)
[Client# pinChat\(\)](#)
[Client# rejectGroupMembershipRequests\(groupId, options\)](#)
[Client# resetState\(\)](#)
[Client# searchMessages\(query\[, options\]\)](#)
[Client# sendMessage\(chatId, content\[, options\]\)](#)
[Client# sendPresenceAvailable\(\)](#)
[Client# sendPresenceUnavailable\(\)](#)
[Client# sendSeen\(chatId\)](#)
[Client# setDisplayName\(displayName\)](#)
[Client# setProfilePicture\(media\)](#)
[Client# setStatus\(status\)](#)
[Client# unarchiveChat\(\)](#)
[Client# unmuteChat\(chatId\)](#)
[Client# unpinChat\(\)](#)

ClientInfo

[ClientInfo\(\)](#)
[ClientInfo# getBatteryStatus\(\)](#)
[ClientInfo# me](#)

[ClientInfo# phone](#)
[ClientInfo# platform](#)
[ClientInfo# pushname](#)

[ClientInfo# wid](#)

Contact

[Contact\(\)](#)
[Contact# block\(\)](#)
[Contact# getAbout\(\)](#)
[Contact# getChat\(\)](#)
[Contact# getCommonGroups\(\)](#)
[Contact# getCountryCode\(\)](#)
[Contact# getFormattedNumber\(\)](#)
[Contact# getProfilePicUrl\(\)](#)

[Contact# id](#)
[Contact# isBlocked](#)
[Contact# isBusiness](#)
[Contact# isEnterprise](#)
[Contact# isGroup](#)
[Contact# isMe](#)
[Contact# isMyContact](#)
[Contact# isUser](#)

[Contact# isWAContact](#)
[Contact# name](#)
[Contact# number](#)

Contact# pushname
Contact# shortName
Contact# unblock()

Events

Events. AUTHENTICATED
Events. AUTHENTICATION_FAILURE
Events. BATTERY_CHANGED
Events. CHAT_ARCHIVED
Events. CHAT_REMOVED
Events. CONTACT_CHANGED
Events. DISCONNECTED
Events. GROUP_ADMIN_CHANGED
Events. GROUP_JOIN

Events. GROUP_LEAVE
Events. GROUP_MEMBERSHIP_REQUEST
Events. GROUP_UPDATE
Events. INCOMING_CALL
Events. LOADING_SCREEN
Events. MEDIA_UPLOADED
Events. MESSAGE_ACK
Events. MESSAGE_CREATE
Events. MESSAGE_EDIT

Events. MESSAGE_REACTION
Events. MESSAGE_RECEIVED
Events. MESSAGE_REVOKED_EVERYONE
Events. MESSAGE_REVOKED_ME
Events. QR_RECEIVED
Events. READY
Events. REMOTE_SESSION_SAVED
Events. STATE_CHANGED
Events. UNREAD_COUNT

GroupChat

GroupChat()
GroupChat# addParticipants(participantIds, options)
GroupChat# approveGroupMembershipRequests(options)
GroupChat# archive()
GroupChat# archived
GroupChat# changeLabels(labelIds)
GroupChat# clearMessages()
GroupChat# clearState()
GroupChat# createdAt
GroupChat# delete()
GroupChat# deletePicture()
GroupChat# demoteParticipants(participantIds)
GroupChat# description
GroupChat# fetchMessages(searchOptions)

[GroupChat# getContact\(\)](#)
[GroupChat# getGroupMembershipRequests\(\)](#)
[GroupChat# getInviteCode\(\)](#)

[GroupChat# getLabels\(\)](#)
[GroupChat# id](#)
[GroupChat# isGroup](#)
[GroupChat# isMuted](#)
[GroupChat# isReadOnly](#)
[GroupChat# lastMessage](#)
[GroupChat# leave\(\)](#)
[GroupChat# markUnread\(\)](#)
[GroupChat# mute\(unmuteDate\)](#)
[GroupChat# muteExpiration](#)
[GroupChat# name](#)
[GroupChat# owner](#)
[GroupChat# participants](#)
[GroupChat# pin\(\)](#)
[GroupChat# pinned](#)
[GroupChat# promoteParticipants\(participantIds\)](#)
[GroupChat# rejectGroupMembershipRequests\(options\)](#)

[GroupChat# removeParticipants\(participantIds\)](#)
[GroupChat# revokeInvite\(\)](#)
[GroupChat# sendMessage\(content\[, options\]\)](#)
[GroupChat# sendSeen\(\)](#)
[GroupChat# sendStateRecording\(\)](#)
[GroupChat# sendStateTyping\(\)](#)
[GroupChat# setDescription\(description\)](#)
[GroupChat# setInfoAdminsOnly\(\[adminsOnly\]\)](#)
[GroupChat# setMessagesAdminsOnly\(\[adminsOnly\]\)](#)
[GroupChat# setPicture\(media\)](#)
[GroupChat# setSubject\(subject\)](#)
[GroupChat# timestamp](#)
[GroupChat# unarchive\(\)](#)
[GroupChat# unmute\(\)](#)
[GroupChat# unpin\(\)](#)
[GroupChat# unreadCount](#)

GroupNotification

[GroupNotification\(\)](#)
[GroupNotification# author](#)
[GroupNotification# body](#)
[GroupNotification# chatId](#)

[GroupNotification# getChat\(\)](#)
[GroupNotification# getContact\(\)](#)
[GroupNotification# getRecipients\(\)](#)
[GroupNotification# id](#)

[GroupNotification# recipientIds](#)
[GroupNotification# reply\(content, options\)](#)

[GroupNotification# timestamp](#)
[GroupNotification# type](#)

GroupNotificationTypes

[GroupNotificationTypes. ADD](#)
[GroupNotificationTypes. ANNOUNCE](#)
[GroupNotificationTypes. DESCRIPTION](#)

[GroupNotificationTypes. INVITE](#)
[GroupNotificationTypes. LEAVE](#)
[GroupNotificationTypes. PICTURE](#)

[GroupNotificationTypes. REMOVE](#)
[GroupNotificationTypes. RESTRICT](#)
[GroupNotificationTypes. SUBJECT](#)

InterfaceController

[InterfaceController\(\)](#)
[InterfaceController# checkFeatureStatus\(feature\)](#)
[InterfaceController# closeRightDrawer\(\)](#)
[InterfaceController# disableFeatures\(features\)](#)

[InterfaceController# enableFeatures\(features\)](#)
[InterfaceController# getFeatures\(\)](#)
[InterfaceController# openChatDrawer\(chatId\)](#)
[InterfaceController# openChatSearch\(chatId\)](#)

[InterfaceController# openChatWindow\(chatId\)](#)
[InterfaceController# openChatWindowAt\(msgId\)](#)
[InterfaceController# openMessageDrawer\(msgId\)](#)

Label

[Label\(client, labelData\)](#)
[Label# getChats\(\)](#)

[Label# hexColor](#)
[Label# id](#)

[Label# name](#)

LegacySessionAuth

[LegacySessionAuth\(options\)](#)

List

[List\(body, buttonText, sections, title, footer\)](#)

[List# _format\(sections\)](#)

[List# buttonText](#)

[List# description](#)

[List# footer](#)

[List# sections](#)

[List# title](#)

LocalAuth

[LocalAuth\(options\)](#)

LocalWebCache

[LocalWebCache\(options\)](#)

Location

[Location\(latitude, longitude\[, options\]\)](#)

[Location# address](#)

[Location# description](#)

[Location# latitude](#)

[Location# longitude](#)

[Location# name](#)

[Location# url](#)

Message

[Message\(\)](#)

[Message# acceptGroupV4Invite\(\)](#)

[Message# ack](#)

[Message# author](#)

[Message# body](#)

[Message# broadcast](#)

[Message# delete\(everyone\)](#)

[Message# deviceType](#)

[Message# downloadMedia\(\)](#)

[Message# duration](#)

[Message# edit\(content\[, options\]\)](#)

[Message# forward\(chat\)](#)

[Message# forwardingScore](#)

Message# from
Message# fromMe
Message# getChat()
Message# getContact()

Message# getInfo()
Message# getMentions()
Message# getOrder()
Message# getPayment()
Message# getQuotedMessage()
Message# getReactions()
Message# hasMedia
Message# hasQuotedMsg
Message# hasReaction
Message# id
Message# inviteV4
Message# isEphemeral
Message# isForwarded
Message# isGif
Message# isStarred
Message# isStatus
Message# links

Message# location
Message# mediaKey
Message# mentionedIds
Message# orderId
Message# rawData
Message# react(reaction)
Message# reload()
Message# reply(content[, chatId][, options])
Message# star()
Message# timestamp
Message# to
Message# token
Message# type
Message# unstar()
Message# vCards

MessageAck

MessageAck. ACK_DEVICE
MessageAck. ACK_ERROR

MessageAck. ACK_PENDING
MessageAck. ACK_PLAYED

MessageAck. ACK_READ
MessageAck. ACK_SERVER

MessageMedia

[MessageMedia\(mimetype, data, filename, filesize\)](#)
[MessageMedia. fromFilePath\(filePath\)](#)
[MessageMedia. fromUrl\(url\[, options\]\)](#)

[MessageMedia# data](#)
[MessageMedia# filename](#)
[MessageMedia# filesize](#)

[MessageMedia# mimetype](#)

MessageTypes

[MessageTypes. AUDIO](#)
[MessageTypes. BROADCAST_NOTIFICATION](#)
[MessageTypes. BUTTONS_RESPONSE](#)
[MessageTypes. CALL_LOG](#)
[MessageTypes. CIPHERTEXT](#)
[MessageTypes. CONTACT_CARD](#)
[MessageTypes. CONTACT_CARD_MULTI](#)
[MessageTypes. DEBUG](#)
[MessageTypes. DOCUMENT](#)
[MessageTypes. E2E_NOTIFICATION](#)
[MessageTypes. GP2](#)
[MessageTypes. GROUP_INVITE](#)

[MessageTypes. GROUP_NOTIFICATION](#)
[MessageTypes. HSM](#)
[MessageTypes. IMAGE](#)
[MessageTypes. INTERACTIVE](#)
[MessageTypes. LIST](#)
[MessageTypes. LIST_RESPONSE](#)
[MessageTypes. LOCATION](#)
[MessageTypes. NATIVE_FLOW](#)
[MessageTypes. NOTIFICATION](#)
[MessageTypes. NOTIFICATION_TEMPLATE](#)
[MessageTypes. ORDER](#)
[MessageTypes. OVERSIZED](#)

[MessageTypes. PAYMENT](#)
[MessageTypes. POLL_CREATION](#)
[MessageTypes. PRODUCT](#)
[MessageTypes. PROTOCOL](#)
[MessageTypes. REACTION](#)
[MessageTypes. REVOKED](#)
[MessageTypes. STICKER](#)
[MessageTypes. TEMPLATE_BUTTON_REPLY](#)
[MessageTypes. TEXT](#)
[MessageTypes. UNKNOWN](#)
[MessageTypes. VIDEO](#)
[MessageTypes. VOICE](#)

NoAuth

[NoAuth\(\)](#)

Order

[Order\(\)](#)

[Order# createdAt](#)

[Order# currency](#)

[Order# subtotal](#)

[Order# total](#)

Payment

[Payment# id](#)

[Payment# paymentAmount1000](#)

[Payment# paymentCurrency](#)

[Payment# paymentMessageReceiverJid](#)

[Payment# paymentNote](#)

[Payment# paymentStatus](#)

[Payment# paymentTransactionTimestamp](#)

[Payment# paymentTxnStatus](#)

Poll

[Poll\(pollName, pollOptions, options\)](#)

[Poll# options](#)

[Poll# pollName](#)

[Poll# pollOptions](#)

PrivateChat

[PrivateChat\(\)](#)

[PrivateChat# archive\(\)](#)

[PrivateChat# archived](#)

[PrivateChat# changeLabels\(labelIds\)](#)

[PrivateChat# clearMessages\(\)](#)

[PrivateChat# clearState\(\)](#)

[PrivateChat# delete\(\)](#)

[PrivateChat# fetchMessages\(searchOptions\)](#)

[PrivateChat# getContact\(\)](#)

[PrivateChat# getLabels\(\)](#)

[PrivateChat# id](#)
[PrivateChat# isGroup](#)
[PrivateChat# isMuted](#)
[PrivateChat# isReadOnly](#)
[PrivateChat# lastMessage](#)
[PrivateChat# markUnread\(\)](#)
[PrivateChat# mute\(unmuteDate\)](#)
[PrivateChat# muteExpiration](#)
[PrivateChat# name](#)
[PrivateChat# pin\(\)](#)

[PrivateChat# pinned](#)
[PrivateChat# sendMessage\(content\[, options\]\)](#)
[PrivateChat# sendSeen\(\)](#)
[PrivateChat# sendStateRecording\(\)](#)
[PrivateChat# sendStateTyping\(\)](#)
[PrivateChat# timestamp](#)
[PrivateChat# unarchive\(\)](#)
[PrivateChat# unmute\(\)](#)
[PrivateChat# unpin\(\)](#)
[PrivateChat# unreadCount](#)

PrivateContact

[PrivateContact\(\)](#)
[PrivateContact# block\(\)](#)
[PrivateContact# getAbout\(\)](#)
[PrivateContact# getChat\(\)](#)
[PrivateContact# getCommonGroups\(\)](#)
[PrivateContact# getCountryCode\(\)](#)
[PrivateContact# getFormattedNumber\(\)](#)
[PrivateContact# getProfilePicUrl\(\)](#)

[PrivateContact# id](#)
[PrivateContact# isBlocked](#)
[PrivateContact# isBusiness](#)
[PrivateContact# isEnterprise](#)
[PrivateContact# isGroup](#)
[PrivateContact# isMe](#)
[PrivateContact# isMyContact](#)
[PrivateContact# isUser](#)

[PrivateContact# isWAContact](#)
[PrivateContact# name](#)
[PrivateContact# number](#)
[PrivateContact# pushname](#)
[PrivateContact# shortName](#)
[PrivateContact# unblock\(\)](#)

Product

[Product\(\)](#)

[Product# currency](#)
[Product# data](#)

[Product# id](#)
[Product# name](#)
[Product# price](#)

[Product# quantity](#)
[Product# thumbnailUrl](#)

ProductMetadata

[ProductMetadata# description](#)
[ProductMetadata# id](#)

[ProductMetadata# name](#)
[ProductMetadata# retailer_id](#)

Reaction

[Reaction\(\)](#)
[Reaction# ack](#)
[Reaction# id](#)
[Reaction# msgId](#)

[Reaction# orphan](#)
[Reaction# orphanReason](#)
[Reaction# reaction](#)
[Reaction# read](#)

[Reaction# senderId](#)
[Reaction# timestamp](#)

RemoteAuth

[RemoteAuth\(options\)](#)

RemoteWebCache

[RemoteWebCache\(options\)](#)

Status

[Status. AUTHENTICATING](#)

[Status. INITIALIZING](#)

[Status. READY](#)

Util

[Util\(\)](#)

[Util. formatImageToWebpSticker\(media\)](#)

[Util. formatToWebpSticker\(media, metadata\)](#)

[Util. formatVideoToWebpSticker\(media\)](#)

[Util. setFfmpegPath\(path\)](#)

WASState

[WASState. CONFLICT](#)

[WASState. CONNECTED](#)

[WASState. DEPRECATED_VERSION](#)

[WASState. OPENING](#)

[WASState. PAIRING](#)

[WASState. PROXYBLOCK](#)

[WASState. SMB_TOS_BLOCK](#)

[WASState. TIMEOUT](#)

[WASState. TOS_BLOCK](#)

[WASState. UNLAUNCHED](#)

[WASState. UNPAIRED](#)

[WASState. UNPAIRED_IDLE](#)

WebCache

[WebCache\(\)](#)

window

[window. compareWebVersions\(lOperand, operator, rOperand\)](#)

[window. injectToFunction\(target, callback\)](#)

[window. WWebJS. generateWaveform\(\)](#)

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.

404

File not found

The site configured at this address does not contain the requested file.

If this is your site, make sure that the filename case matches the URL as well as any file permissions.

For root URLs (like <http://example.com/>) you must provide an index.html file.

[Read the full documentation](#) for more information about using **GitHub Pages** .

[GitHub Status](#) – [@githubstatus](#)



[whatsapp-web. js 1. 23. 0](#)
class

Base

Source: [structures/ Base. js:6](#)

Represents a WhatsApp data structure

new Base ()

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.

[whatsapp-web. js 1. 23. 0](#)
class

BaseAuthStrategy

Source: [authStrategies/ BaseAuthStrategy. js:6](#)

Base class which all authentication strategies extend

new BaseAuthStrategy ()

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.

[whatsapp-web. js 1. 23. 0](#)
class

BusinessContact

Source: [structures/ BusinessContact. js:9](#)

Represents a Business Contact on WhatsApp

Properties

[businessProfile](#)

[id](#)

[isBlocked](#)

[isBusiness](#)

[isEnterprise](#)

[isGroup](#)

[isMe](#)

[isMyContact](#)

[isUser](#)

[isWABusiness](#)

[name](#)

[number](#)

[pushname](#)

[shortName](#)

Methods

[block\(\)](#)

[getAbout\(\)](#)

[getChat\(\)](#)

[getCommonGroups\(\)](#)

[getCountryCode\(\)](#)

[getFormattedNumber\(\)](#)

[getProfilePicUrl\(\)](#)

[unblock\(\)](#)

new BusinessContact ()

Extends

[Contact](#)

Properties

businessProfile

The contact's business profile

id unknown

ID that represents the contact

Inherited from

[Contact#id](#)

isBlocked unknown

Indicates if you have blocked this contact

Inherited from

[Contact#isBlocked](#)

isBusiness unknown

Indicates if the contact is a business contact

Inherited from

[Contact#isBusiness](#)

isEnterprise unknown

Indicates if the contact is an enterprise contact

Inherited from

[Contact#isEnterprise](#)

isGroup unknown

Indicates if the contact is a group contact

Inherited from

[Contact#isGroup](#)

isMe unknown

Indicates if the contact is the current user's contact

Inherited from

[Contact#isMe](#)

isMyContact unknown

Indicates if the number is saved in the current phone's contacts

Inherited from

[Contact#isMyContact](#)

isUser unknown

Indicates if the contact is a user contact

Inherited from

[Contact#isUser](#)

isWAContact unknown

Indicates if the number is registered on WhatsApp

Inherited from
[Contact#isWAContact](#)

name unknown

The contact's name, as saved by the current user

Inherited from
[Contact#name](#)

number unknown

Contact's phone number

Inherited from
[Contact#number](#)

pushname unknown

The name that the contact has configured to be shown publically

Inherited from
[Contact#pushname](#)

shortName unknown

A shortened version of name

Inherited from
[Contact#shortName](#)

Methods

async

block () → Promise containing boolean

Blocks this contact from WhatsApp

Inherited from
[Contact#block](#)

Returns

async

getAbout () → Promise containing nullable string

Gets the Contact's current "about" info. Returns null if you don't have permission to read their status.

Inherited from
[Contact#getAbout](#)

Returns

async

getChat () → Promise containing [Chat](#)

Returns the Chat that corresponds to this Contact. Will return null when getting chat for currently logged in user.

Inherited from

[Contact#getChat](#)

Returns

async

getCommonGroups () → Promise containing Array of WAWebJS.ChatId

Gets the Contact's common groups with you. Returns empty array if you don't have any common group.

Inherited from

[Contact#getCommonGroups](#)

Returns

async

getCountryCode () → Promise containing string

Returns the contact's countrycode, ([\[email protected\]](#)) => (1)

Inherited from

[Contact#getCountryCode](#)

Returns

async

getFormattedNumber () → Promise containing string

Returns the contact's formatted phone number, ([\[email protected\]](#)) => (+1 (234) 5678-901)

Inherited from

[Contact#getFormattedNumber](#)

Returns

async

getProfilePicUrl () → Promise containing string

Returns the contact's profile picture URL, if privacy settings allow it

Inherited from

[Contact#getProfilePicUrl](#)

Returns

async

unlock () → Promise containing boolean

Unlocks this contact from WhatsApp

Inherited from

[Contact#unlock](#)

Returns

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.

[whatsapp-web.js 1.23.0](#)
class

Buttons

Source: [structures/ Buttons.js:23](#)

Message type buttons

Properties

[body](#)

[buttons](#)

[footer](#)

[title](#)

Method

[_format\(buttons\)](#)

new Buttons (body, buttons, title, footer)

Parameters

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

body			
------	--	--	--

buttons		See	ButtonSpec
---------	--	-----	----------------------------

Name	Type	Optional	Description
title			Value can be null.
footer			Value can be null.

Properties

body (string or [MessageMedia](#))

Message body

buttons Array of [FormattedButtonSpec](#)

buttons of message

footer string

footer of message

title string

title of message

Method

_format (buttons) → Array of [FormattedButtonSpec](#)

Creates button array from simple array

Example

Input: [{id: 'customId', body: 'button1'}, {body: 'button2'}, {body: 'button3'}, {body: 'button4'}]

Returns: [{ buttonId: 'customId', buttonText: { 'displayText': 'button1', type: 1 }, {buttonId:

Parameter

Name	Type	Optional	Description
buttons	Array of ButtonSpec		

Returns

Array of [FormattedButtonSpec](#)

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.

[whatsapp-web.js 1.23.0](#)
class

Call

Source: [structures/ Call.js:9](#)

Represents a Call on WhatsApp

Properties

[canHandleLocally](#)
[from](#)
[fromMe](#)

[id](#)
[isGroup](#)
[isVideo](#)

[participants](#)
[timestamp](#)
[webClientShouldHandle](#)

Method

[reject\(\)](#)

new Call ()

Extends
[Base](#)

Properties

canHandleLocally boolean

Indicates if the call can be handled in waweb

from string

From

fromMe boolean

Indicates if the call was sent by the current user

id string

Call ID

isGroup boolean

Is Group

isVideo boolean

Is video

participants object

Object with participants

timestamp number

Unix timestamp for when the call was created

webClientShouldHandle boolean

Indicates if the call Should be handled in waweb

Method

async

reject ()

Reject the call

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.

[whatsapp-web.js 1.23.0](#)
class

Chat

Source: [structures/ Chat.js:10](#)

Represents a Chat on WhatsApp

Properties

[archived](#)
[id](#)
[isGroup](#)
[isMuted](#)

[isReadOnly](#)
[lastMessage](#)
[muteExpiration](#)
[name](#)

[pinned](#)
[timestamp](#)
[unreadCount](#)

Methods

[archive\(\)](#)
[changeLabels\(labelIds\)](#)
[clearMessages\(\)](#)
[clearState\(\)](#)
[delete\(\)](#)
[fetchMessages\(searchOptions\)](#)

[getContact\(\)](#)
[getLabels\(\)](#)
[markUnread\(\)](#)
[mute\(unmuteDate\)](#)
[pin\(\)](#)
[sendMessage\(content\[, options\]\)](#)

[sendSeen\(\)](#)
[sendStateRecording\(\)](#)
[sendStateTyping\(\)](#)
[unarchive\(\)](#)
[unmute\(\)](#)
[unpin\(\)](#)

new Chat ()

Extends
[Base](#)

Properties

archived boolean

Indicates if the Chat is archived

id object

ID that represents the chat

isGroup boolean

Indicates if the Chat is a Group Chat

isMuted boolean

Indicates if the chat is muted or not

isReadOnly boolean

Indicates if the Chat is readonly

lastMessage [Message](#)

Last message fo chat

muteExpiration number

Unix timestamp for when the mute expires

name string

Title of the chat

pinned boolean

Indicates if the Chat is pinned

timestamp number

Unix timestamp for when the last activity occurred

unreadCount number

Amount of messages unread

Methods

async

archive ()

Archives this chat

async

changeLabels (labelIds) → Promise containing void

Add or remove labels to this Chat

Parameter

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

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

labelIds	Array of (number or string)	
----------	-----------------------------	--

Returns

Promise containing void

async

clearMessages () → Promise containing Boolean

Clears all messages from the chat

Returns

Promise containing Boolean

result

async

clearState ()

Stops typing or recording in chat immediately.

async

delete () → Promise containing Boolean

Deletes the chat

Returns

Promise containing Boolean

result

async

fetchMessages (searchOptions) → Promise containing Array of [Message](#)

Loads chat messages, sorted from earliest to latest.

Parameters

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

searchOptions	Object	Options for searching messages. Right now only limit and fromMe is supported.
---------------	--------	---

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

Values in searchOptions have the following properties:

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

limit	Number	Yes	The amount of messages to return. If no limit is specified, the available messages will be returned. Note that the actual number of returned messages may be smaller if there aren't enough messages in the conversation. Set this to Infinity to load all messages.
-------	--------	-----	--

fromMe	Boolean	Yes	Return only messages from the bot number or vice versa. To get all messages, leave the option undefined.
--------	---------	-----	--

Returns

Promise containing Array of [Message](#)

async

getContact () → Promise containing [Contact](#)

Returns the Contact that corresponds to this Chat.

Returns

Promise containing [Contact](#)

async

getLabels () → Promise containing Array of [Label](#)

Returns array of all Labels assigned to this Chat

Returns

Promise containing Array of [Label](#)

async

markUnread ()

Mark this chat as unread

async

mute (unmuteDate)

Mutes this chat forever, unless a date is specified

Parameter

Name	Type	Optional	Description
unmuteDate	Date		Date at which the Chat will be unmuted, leave as is to mute forever
			Value can be null.

async

pin () → Promise containing boolean

Pins this chat

Returns

Promise containing boolean

New pin state. Could be false if the max number of pinned chats was reached.

async

sendMessage (content[, options]) → Promise containing [Message](#)

Send a message to this chat

Parameters

Name	Type	Optional	Description
content	(string, MessageMedia , or Location)		
options	MessageSendOptions	Yes	

Returns

Promise containing [Message](#)

Message that was just sent

async

sendSeen () → Promise containing Boolean

Set the message as seen

Returns

Promise containing Boolean

result

async

sendStateRecording ()

Simulate recording audio in chat. This will last for 25 seconds.

async

sendStateTyping ()

Simulate typing in chat. This will last for 25 seconds.

async

unarchive ()

un-archives this chat

async

unmute ()

Unmutes this chat

async

unpin () → Promise containing boolean

Unpins this chat

Returns

Promise containing boolean

New pin state

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.

[whatsapp-web.js 1.23.0](#)
class

Client

Source: [Client.js:56](#)

Starting point for interacting with the WhatsApp Web API

Property

[info](#)

Methods

[acceptGroupV4Invite\(inviteInfo\)](#)
[acceptInvite\(inviteCode\)](#)
[addOrRemoveLabels\(labelIds, chatIds\)](#)
[approveGroupMembershipRequests\(groupId, options\)](#)
[archiveChat\(\)](#)
[createGroup\(title, participants, options\)](#)
[deleteProfilePicture\(\)](#)
[destroy\(\)](#)
[getBlockedContacts\(\)](#)
[getChatById\(chatId\)](#)
[getChatLabels\(chatId\)](#)
[getChats\(\)](#)
[getChatsByLabelId\(labelId\)](#)
[getCommonGroups\(contactId\)](#)
[getContactById\(contactId\)](#)

[getContacts\(\)](#)
[getCountryCode\(number\)](#)

[getFormattedNumber\(number\)](#)
[getGroupMembershipRequests\(groupId\)](#)
[getInviteInfo\(inviteCode\)](#)
[getLabelById\(labelId\)](#)
[getLabels\(\)](#)
[getNumberId\(number\)](#)
[getProfilePicUrl\(contactId\)](#)
[getState\(\)](#)
[getWebVersion\(\)](#)
[initialize\(\)](#)
[isRegisteredUser\(id\)](#)
[logout\(\)](#)
[markChatUnread\(chatId\)](#)

[muteChat\(chatId, unmuteDate\)](#)
[pinChat\(\)](#)
[rejectGroupMembershipRequests\(groupId, options\)](#)
[resetState\(\)](#)
[searchMessages\(query\[, options\]\)](#)
[sendMessage\(chatId, content\[, options\]\)](#)
[sendPresenceAvailable\(\)](#)
[sendPresenceUnavailable\(\)](#)
[sendSeen\(chatId\)](#)
[setDisplayName\(displayName\)](#)
[setProfilePicture\(media\)](#)
[setStatus\(status\)](#)
[unarchiveChat\(\)](#)
[unmuteChat\(chatId\)](#)
[unpinChat\(\)](#)

Events

[auth_failure](#)
[authenticated](#)
[change_battery](#)
[change_state](#)
[chat_archived](#)
[chat_removed](#)
[contact_changed](#)
[disconnected](#)

[group_admin_changed](#)
[group_join](#)
[group_leave](#)
[group_membership_request](#)
[group_update](#)
[incoming_call](#)
[media_uploaded](#)
[message](#)

[message_ack](#)
[message_create](#)
[message_edit](#)

[message_reaction](#)
[message_revoke_everyone](#)
[message_revoke_me](#)
[qr](#)
[ready](#)

new Client (options)

Parameters

Name	Type	Optional	Description
			Client options
			Values in options have the following properties:
Name	Type	Optional	Description
			Determines how to save and restore sessions. Will use LegacySessionAuth if options.session is set. Otherwise, NoAuth will be used.
authStrategy			
			The version of WhatsApp Web to use. Use options.webVersionCache to configure how the version is retrieved.
webVersion			
options			
			Determines how to retrieve the WhatsApp Web version. Defaults to a local cache (LocalWebCache) that falls back to latest the requested version is not found.
webVersionCache			
			Timeout for authentication select in puppeteer
authTimeoutMs			
			Puppeteer launch options. View docs here: https://
puppeteer			

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

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

github.com/puppeteer/
puppeteer/

qrMaxRetries

How many times should
the qrcode be refresh
before giving up

restartOnAuthFail

@deprecated This opti
should be set directl
on the
LegacySessionAuth.

session

@deprecated Only here
for backwards-
compatibility. You
should move to using
LocalAuth, or set the
authStrategy to
LegacySessionAuth
explicitly.

takeoverOnConflict

If another whatsapp w
session is detected
(another browser), ta
over the session in t
current browser

takeoverTimeoutMs

How much time to wait
before taking over th
session

userAgent

User agent to use in
puppeteer

ffmpegPath

Ffmpeg path to use wh
formatting videos to
webp while sending
stickers

bypassCSP

Sets bypassing of
page's Content-
Security-Policy.

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

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

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

proxyAuthentication			
---------------------	--	--	--

Proxy Authentication object.

Extends

EventEmitter

Fires

[Client#event:qr](#)

[Client#event:authenticated](#)

[Client#event:auth_failure](#)

[Client#event:ready](#)

[Client#event:message](#)

[Client#event:message_ack](#)

[Client#event:message_create](#)

[Client#event:message_revoke_me](#)

[Client#event:message_revoke_everyone](#)

[Client#event:media_uploaded](#)

[Client#event:group_join](#)

[Client#event:group_leave](#)

[Client#event:group_update](#)

[Client#event:disconnected](#)

[Client#event:change_state](#)

[Client#event:contact_changed](#)

[Client#event:group_admin_changed](#)

[Client#event:group_membership_request](#)

Property

info [ClientInfo](#)

Current connection information

Methods

async

acceptGroupV4Invite (inviteInfo) → Promise containing Object

Accepts a private invitation to join a group

Parameter

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

inviteInfo	object		
------------	--------	--	--

Invite V4 Info

Returns

Promise containing Object

async

acceptInvite (inviteCode) → Promise containing string

Accepts an invitation to join a group

Parameter

Name	Type	Optional Description
inviteCode	string	Invitation code

Returns

Promise containing string

Id of the joined Chat

async

addOrRemoveLabels (labelIds, chatIds) → Promise containing void

Change labels in chats

Parameters

Name	Type	Optional Description
labelIds	Array of (number or string)	
chatIds	Array of string	

Returns

Promise containing void

async

approveGroupMembershipRequests (groupId, options) → Promise containing Array of [MembershipRequestActionResult](#)

Approves membership requests if any

Parameters

Name	Type	Optional Description
groupId	string	The group ID to get the membership request for
options	MembershipRequestActionOptions	Options for performing a membership request action

Returns

Promise containing Array of [MembershipRequestActionResult](#)

Returns an array of requester IDs whose membership requests were approved and an error for each requester, if any occurred during the operation. If there are no requests, an empty array will be returned

async

archiveChat () → boolean

Enables and returns the archive state of the Chat

Returns

boolean

async

createGroup (title, participants, options) → Promise containing ([CreateGroupResult](#) or string)

Creates a new group

Parameters

Name	Type	Optional Description
title	string	Group title
participants	(string, Contact , Array of (Contact or string), or undefined)	A single Contact object or an ID as a string or an array of Contact objects

Name	Type	Optional Description
		or contact IDs to add to the group
options	CreateGroupOptions	An object that handles options for group creation

Returns

Promise containing ([CreateGroupResult](#) or string)

Object with resulting data or an error message as a string

async

deleteProfilePicture () → Promise containing boolean

Deletes the current user's profile picture.

Returns

Promise containing boolean

Returns true if the picture was properly deleted.

async

destroy ()

Closes the client

async

getBlockedContacts () → Promise containing Array of [Contact](#)

Gets all blocked contacts by host account

Returns

Promise containing Array of [Contact](#)

async

getChatById (chatId) → Promise containing [Chat](#)

Get chat instance by ID

Parameter

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

chatId	string		
--------	--------	--	--

Returns

Promise containing [Chat](#)

async

getChatLabels (chatId) → Promise containing Array of [Label](#)

Get all Labels assigned to a chat

Parameter

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

chatId	string		
--------	--------	--	--

Returns

Promise containing Array of [Label](#)

async

getChats () → Promise containing Array of [Chat](#)

Get all current chat instances

Returns

Promise containing Array of [Chat](#)

async

getChatsByLabelId (labelId) → Promise containing Array of [Chat](#)

Get all Chats for a specific Label

Parameter

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

labelId	string		
---------	--------	--	--

Returns

Promise containing Array of [Chat](#)

async

getCommonGroups (contactId) → Promise containing Array of WAWebJS.ChatId

Gets the Contact's common groups with you. Returns empty array if you don't have any common group.

Parameter

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

contactId	string		the whatsapp user's ID (_serialized format)
-----------	--------	--	---

Returns

Promise containing Array of WAWebJS.ChatId

async

getContactById (contactId) → Promise containing [Contact](#)

Get contact instance by ID

Parameter

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

contactId	string		
-----------	--------	--	--

Returns

Promise containing [Contact](#)

async

getContacts () → Promise containing Array of [Contact](#)

Get all current contact instances

Returns

Promise containing Array of [Contact](#)

async

getCountryCode (number) → Promise containing string

Get the country code of a WhatsApp ID.

Parameter

Name	Type	Optional	Description
number	string		Number or ID

Returns

Promise containing string

async

getFormattedNumber (number) → Promise containing string

Get the formatted number of a WhatsApp ID.

Parameter

Name	Type	Optional	Description
number	string		Number or ID

Returns

Promise containing string

async

getGroupMembershipRequests (groupId) → Promise containing Array of [GroupMembershipRequest](#)

Gets an array of membership requests

Parameter

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

groupId	string		The ID of a group to get membership requests for
---------	--------	--	--

Returns

Promise containing Array of [GroupMembershipRequest](#)

An array of membership requests

async

getInviteInfo (inviteCode) → Promise containing object

Returns an object with information about the invite code's group

Parameter

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

inviteCode	string		
------------	--------	--	--

Returns

Promise containing object

Invite information

async

getLabelById (labelId) → Promise containing [Label](#)

Get Label instance by ID

Parameter

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

labelId	string		
---------	--------	--	--

Returns

Promise containing [Label](#)

async

getLabels () → Promise containing Array of [Label](#)

Get all current Labels

Returns

Promise containing Array of [Label](#)

async

getNumberId (number) → Promise containing (Object or null)

Get the registered WhatsApp ID for a number. Will return null if the number is not registered on WhatsApp.

Parameter

Name	Type	Optional	Description
number	string		Number or ID ("@c.us" will be automatically appended if not specified)

Returns

Promise containing (Object or null)

async

getProfilePicUrl (contactId) → Promise containing string

Returns the contact ID's profile picture URL, if privacy settings allow it

Parameter

Name	Type	Optional	Description
contactId	string		the whatsapp user's ID

Returns

Promise containing string

async

getState () → [WAState](#)

Gets the current connection state for the client

Returns

[WASState](#)

async

getWebVersion () → Promise containing string

Returns the version of WhatsApp Web currently being run

Returns

Promise containing string

async

initialize ()

Sets up events and requirements, kicks off authentication request

async

isRegisteredUser (id) → Promise containing Boolean

Check if a given ID is registered in whatsapp

Parameter

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

id	string		the whatsapp user's ID
----	--------	--	------------------------

Returns

Promise containing Boolean

async

logout ()

Logs out the client, closing the current session

async

markChatUnread (chatId)

Mark the Chat as unread

Parameter

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

chatId	string		ID of the chat that will be marked as unread
--------	--------	--	--

async

muteChat (chatId, unmuteDate)

Mutes this chat forever, unless a date is specified

Parameters

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

chatId	string		ID of the chat that will be muted
--------	--------	--	-----------------------------------

unmuteDate	Date		Date when the chat will be unmuted, leave as is to mute forever
------------	------	--	---

Value can be null.

async

pinChat () → Promise containing boolean

Pins the Chat

Returns

Promise containing boolean

New pin state. Could be false if the max number of pinned chats was reached.

async

rejectGroupMembershipRequests (groupId, options) → Promise containing Array of [MembershipRequestActionResult](#)

Rejects membership requests if any

Parameters

Name	Type	Optional	Description
groupId	string		The group ID to get the membership request for
options	MembershipRequestActionOptions		Options for performing a membership request action

Returns

Promise containing Array of [MembershipRequestActionResult](#)

Returns an array of requester IDs whose membership requests were rejected and an error for each requester, if any occurred during the operation. If there are no requests, an empty array will be returned

async

resetState ()

Force reset of connection state for the client

async

searchMessages (query[, options]) → Promise containing Array of [Message](#)

Searches for messages

Parameters

Name	Type	Optional	Description
query	string		
Values in options have the following properties:			
options	Object	Yes	
page	number	Yes	

Name	Type	Optional	Description
Name	Type	Optional	Description
limit	number	Yes	
chatId	string	Yes	

Returns

Promise containing Array of [Message](#)

async

sendMessage (chatId, content[, options]) → Promise containing [Message](#)

Send a message to a specific chatId

Parameters

Name	Type	Optional	Description
chatId	string		
content	(string, MessageMedia , Location , Poll , Contact , Array of Contact , Buttons , or List)		
options	MessageSendOptions	Yes	Options used when sending the message

Returns

Promise containing [Message](#)

Message that was just sent

async

sendPresenceAvailable ()

Marks the client as online

async

sendPresenceUnavailable ()

Marks the client as unavailable

async

sendSeen (chatId) → Promise containing boolean

Mark as seen for the Chat

Parameter

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

chatId	string		
--------	--------	--	--

Returns

Promise containing boolean

result

async

setDisplayName (displayName) → Promise containing Boolean

Sets the current user's display name. This is the name shown to WhatsApp users that have not added you as a contact beside your number in groups and in your profile.

Parameter

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

displayName	string		New display name
-------------	--------	--	------------------

Returns

Promise containing Boolean

async

setProfilePicture (media) → Promise containing boolean

Sets the current user's profile picture.

Parameter

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

media	MessageMedia		
-------	------------------------------	--	--

Returns

Promise containing boolean

Returns true if the picture was properly updated.

async

setStatus (status)

Sets the current user's status message

Parameter

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

status	string		New status message
--------	--------	--	--------------------

async

unarchiveChat () → boolean

Changes and returns the archive state of the Chat

Returns

boolean

async

unmuteChat (chatId)

Unmutes the Chat

Parameter

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

chatId	string		ID of the chat that will be unmuted
--------	--------	--	-------------------------------------

async

unpinChat () → Promise containing boolean

Unpins the Chat

Returns

Promise containing boolean

New pin state

Events

auth_failure

Emitted when there has been an error while trying to restore an existing session

Parameter

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

message	string		
---------	--------	--	--

authenticated

Emitted when authentication is successful

change_battery

Emitted when the battery percentage for the attached device changes. Will not be sent if using multi-device.

Parameters

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

Values in batteryInfo have the following properties:

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

batteryInfo object	battery number		The current battery percentage
--------------------	----------------	--	--------------------------------

plugged	boolean		Indicates if the phone is plugged in
---------	---------	--	--------------------------------------

Name	Type	Optional Description
Name	Type	Optional Description (true) or not (false)

Deprecated

change_state

Emitted when the connection state changes

Parameter

Name	Type	Optional Description
state	WASState	the new connection state

chat_archived

Emitted when a chat is archived/unarchived

Parameters

Name	Type	Optional Description
chat	Chat	

currState boolean

prevState boolean

chat_removed

Emitted when a chat is removed

Parameter

Name	Type	Optional Description
chat	Chat	

contact_changed

Emitted when a contact or a group participant changes their phone number.

Parameters

Name	Type	Optional	Description
message	Message		Message with more information about the event.
oldId	String		The user's id (an old one) who changed their phone number and who triggered the notification.
newId	String		The user's new id after the change.
isContact	Boolean		Indicates if a contact or a group participant changed their phone number.

disconnected

Emitted when the client has been disconnected

Parameter

Name	Type	Optional	Description
reason	(WASState or "NAVIGATION")		reason that caused the disconnect

group_admin_changed

Emitted when a current user is promoted to an admin or demoted to a regular user.

Parameter

Name	Type	Optional	Description
notification	GroupNotification		GroupNotification with more information about the action

group_join

Emitted when a user joins the chat via invite link or is added by an admin.

Parameter

Name	Type	Optional	Description
notification	GroupNotification		GroupNotification with more information about the action

group_leave

Emitted when a user leaves the chat or is removed by an admin.

Parameter

Name	Type	Optional	Description
notification	GroupNotification		GroupNotification with more information about the action

group_membership_request

Emitted when some user requested to join the group that has the membership approval mode turned on

Parameters

Name	Type	Optional	Description						
notification	GroupNotification		GroupNotification with more information about the action						
			Values in notification have the following properties:						
		<table><tr><th>Name</th><th>Type</th><th>Optional</th><th>Description</th></tr><tr><td>chatId</td><td>string</td><td></td><td>The group ID the request was made for</td></tr></table>	Name	Type	Optional	Description	chatId	string	
Name	Type	Optional	Description						
chatId	string		The group ID the request was made for						

Name	Type	Optional Description									
		<table> <tr> <th>Name</th><th>Type</th><th>Optional Description</th></tr> <tr> <td>author</td><td>string</td><td>The user ID that made a request</td></tr> <tr> <td>timestamp</td><td>number</td><td>The timestamp the request was made at</td></tr> </table>	Name	Type	Optional Description	author	string	The user ID that made a request	timestamp	number	The timestamp the request was made at
Name	Type	Optional Description									
author	string	The user ID that made a request									
timestamp	number	The timestamp the request was made at									

group_update

Emitted when group settings are updated, such as subject, description or picture.

Parameter

Name	Type	Optional Description
notification	GroupNotification	GroupNotification with more information about the action

incoming_call

Emitted when a call is received

Parameters

Name	Type	Optional Description													
Values in call have the following properties:															
		<table> <tr> <th>Name</th><th>Type</th><th>Optional Description</th></tr> <tr> <td>id</td><td>number</td><td>Call id</td></tr> <tr> <td rowspan="3">call object</td><td>peerJid</td><td>Who called</td></tr> <tr> <td>isVideo</td><td>if is video</td></tr> <tr> <td>isGroup</td><td>if is group</td></tr> </table>	Name	Type	Optional Description	id	number	Call id	call object	peerJid	Who called	isVideo	if is video	isGroup	if is group
Name	Type	Optional Description													
id	number	Call id													
call object	peerJid	Who called													
	isVideo	if is video													
	isGroup	if is group													

Name	Type	Optional Description	Name	Type	Optional Description
			canHandleLocally	boolean	if we can handle in waweb
			outgoing	boolean	if is outgoing
			webClientShouldHandle	boolean	If Waweb should handle
			participants	object	Participants

media_uploaded

Emitted when media has been uploaded for a message sent by the client.

Parameter

Name	Type	Optional Description
message	Message	The message with media that was uploaded

message

Emitted when a new message is received.

Parameter

Name	Type	Optional Description
message	Message	The message that was received

message_ack

Emitted when an ack event occurs on message type.

Parameters

Name	Type	Optional Description
message	Message	The message that was affected
ack	MessageAck	The new ACK value

message_create

Emitted when a new message is created, which may include the current user's own messages.

Parameter

Name	Type	Optional Description
message	Message	The message that was created

message_edit

Emitted when messages are edited

Parameters

Name	Type	Optional Description
message	Message	
newBody	string	
prevBody	string	

message_reaction

Emitted when a reaction is sent, received, updated or removed

Parameters

Name	Type	Optional Description
reaction	object	Values in reaction have the following properties:

Name	Type	Optional	Description
Name	Type	Optional	Description
id	object		Reaction id
orphan	number		Orphan
orphanReason	string		Orphan reason Value can be null.
timestamp	number		Timestamp
reaction	string		Reaction
read	boolean		Read
msgId	object		Parent message id
senderId	string		Sender id
ack	number		Ack Value can be null.

message_revoke_everyone

Emitted when a message is deleted for everyone in the chat.

Parameters

Name	Type	Optional	Description
message	Message		The message that was revoked, in its current state. It will not contain the original message's data.

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

revoked_msg	Message		The message that was revoked, before it was revoked. It will contain the message's original data. Note that due to the way this data is captured, it may be possible that this param will be undefined. Value can be null.
-------------	-------------------------	--	---

message_revoke_me

Emitted when a message is deleted by the current user.

Parameter

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

message	Message		The message that was revoked
---------	-------------------------	--	------------------------------

qr

Emitted when a QR code is received

Parameter

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

qr	string		QR Code
----	--------	--	---------

ready

Emitted when the client has initialized and is ready to receive messages.

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.

[whatsapp-web.js 1.23.0](#)
class

ClientInfo

Source: [structures/ClientInfo.js:9](#)

Current connection information

Properties

[me](#)
[phone](#)

[platform](#)
[pushname](#)

[wid](#)

Method

[getBatteryStatus\(\)](#)

new ClientInfo ()

Extends
[Base](#)

Properties

me object

Deprecated
Use .wid instead

phone object

Information about the phone this client is connected to. Not available in multi-device.

Properties

Name	Type	Optional Description
wa_version	string	WhatsApp Version running on the phone
os_version	string	OS Version running on the phone (iOS or Android version)
device_manufacturer	string	Device manufacturer
device_model	string	Device model

Name	Type	Optional Description
os_build_number	string	OS build number

Deprecated

platform string

Platform WhatsApp is running on

pushname string

Name configured to be shown in push notifications

wid object

Current user ID

Method

async

getBatteryStatus () → (object, number, or boolean)

Get current battery percentage and charging status for the attached device

Deprecated
Returns

object

batteryStatus

number

batteryStatus.battery - The current battery percentage

boolean

batteryStatus.plugged - Indicates if the phone is plugged in (true) or not (false)

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.

[whatsapp-web.js 1.23.0](#)
class

Contact

Source: [structures/ Contact. js:17](#)

Represents a Contact on WhatsApp

Properties

[id](#)
[isBlocked](#)
[isBusiness](#)
[isEnterprise](#)
[isGroup](#)

[isMe](#)
[isMyContact](#)
[isUser](#)
[isWAContact](#)
[name](#)

[number](#)
[pushname](#)
[shortName](#)

Methods

[block\(\)](#)
[getAbout\(\)](#)
[getChat\(\)](#)

[getCommonGroups\(\)](#)
[getCountryCode\(\)](#)
[getFormattedNumber\(\)](#)

[getProfilePicUrl\(\)](#)
[unblock\(\)](#)

new Contact ()

Extends
[Base](#)

Properties

id [ContactId](#)

ID that represents the contact

isBlocked boolean

Indicates if you have blocked this contact

isBusiness boolean

Indicates if the contact is a business contact

isEnterprise boolean

Indicates if the contact is an enterprise contact

isGroup boolean

Indicates if the contact is a group contact

isMe boolean

Indicates if the contact is the current user's contact

isMyContact boolean

Indicates if the number is saved in the current phone's contacts

isUser boolean

Indicates if the contact is a user contact

isWAContact boolean

Indicates if the number is registered on WhatsApp

name nullable string

The contact's name, as saved by the current user

number string

Contact's phone number

pushname string

The name that the contact has configured to be shown publically

shortName nullable string

A shortened version of name

Methods

async

block () → Promise containing boolean

Blocks this contact from WhatsApp

Returns

Promise containing boolean

async

getAbout () → Promise containing nullable string

Gets the Contact's current "about" info. Returns null if you don't have permission to read their status.

Returns

Promise containing nullable string

async

getChat () → Promise containing [Chat](#)

Returns the Chat that corresponds to this Contact. Will return null when getting chat for currently logged in user.

Returns

Promise containing [Chat](#)

async

getCommonGroups () → Promise containing Array of WAWebJS.ChatId

Gets the Contact's common groups with you. Returns empty array if you don't have any common group.

Returns

Promise containing Array of WAWebJS.ChatId

async

getCountryCode () → Promise containing string

Returns the contact's countrycode, ([\[email protected\]](#)) => (1)

Returns

Promise containing string

async

getFormattedNumber () → Promise containing string

Returns the contact's formatted phone number, (
[\[email protected\]](#)) => (+1 (234) 5678-901)

Returns

Promise containing string

async

getProfilePicUrl () → Promise containing string

Returns the contact's profile picture URL, if privacy settings allow it

Returns

Promise containing string

async

unblock () → Promise containing boolean

Unblocks this contact from WhatsApp

Returns

Promise containing boolean

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.

[whatsapp-web. js 1. 23. 0](#)
class

GroupChat

Source: [structures/ GroupChat. js:17](#)

Represents a Group Chat on WhatsApp

Properties

[archived](#)
[createdAt](#)
[description](#)
[id](#)
[isGroup](#)

[isMuted](#)
[isReadOnly](#)
[lastMessage](#)
[muteExpiration](#)
[name](#)

[owner](#)
[participants](#)
[pinned](#)
[timestamp](#)
[unreadCount](#)

Methods

[addParticipants\(participantIds, options\)](#)
[approveGroupMembershipRequests\(options\)](#)
[archive\(\)](#)
[changeLabels\(labelIds\)](#)
[clearMessages\(\)](#)
[clearState\(\)](#)
[delete\(\)](#)
[deletePicture\(\)](#)
[demoteParticipants\(participantIds\)](#)
[fetchMessages\(searchOptions\)](#)
[getContact\(\)](#)
[getGroupMembershipRequests\(\)](#)

[getInviteCode\(\)](#)
[getLabels\(\)](#)
[leave\(\)](#)
[markUnread\(\)](#)
[mute\(unmuteDate\)](#)
[pin\(\)](#)
[promoteParticipants\(participantIds\)](#)
[rejectGroupMembershipRequests\(options\)](#)
[removeParticipants\(participantIds\)](#)
[revokeInvite\(\)](#)
[sendMessage\(content\[, options\]\)](#)
[sendSeen\(\)](#)

[sendStateRecording\(\)](#)
[sendStateTyping\(\)](#)
[setDescription\(description\)](#)
[setInfoAdminsOnly\(\[adminsOnly\]\)](#)
[setMessagesAdminsOnly\(\[adminsOnly\]\)](#)
[setPicture\(media\)](#)
[setSubject\(subject\)](#)
[unarchive\(\)](#)
[unmute\(\)](#)
[unpin\(\)](#)

new GroupChat ()

Extends

[Chat](#)

Properties

archived unknown

Indicates if the Chat is archived

Inherited from

[Chat#archived](#)

createdAt date

Gets the date at which the group was created

description string

Gets the group description

id unknown

ID that represents the chat

Inherited from

[Chat#id](#)

isGroup unknown

Indicates if the Chat is a Group Chat

Inherited from

[Chat#isGroup](#)

isMuted unknown

Indicates if the chat is muted or not

Inherited from

[Chat#isMuted](#)

isReadOnly unknown

Indicates if the Chat is readonly

Inherited from

[Chat#isReadOnly](#)

lastMessage unknown

Last message fo chat

Inherited from

[Chat#lastMessage](#)

muteExpiration unknown

Unix timestamp for when the mute expires

Inherited from

[Chat#muteExpiration](#)

name unknown

Title of the chat

Inherited from

[Chat#name](#)

owner [ContactId](#)

Gets the group owner

participants Array of [GroupParticipant](#)

Gets the group participants

pinned unknown

Indicates if the Chat is pinned

Inherited from

[Chat#pinned](#)

timestamp unknown

Unix timestamp for when the last activity occurred

Inherited from

[Chat#timestamp](#)

unreadCount unknown

Amount of messages unread

Inherited from

[Chat#unreadCount](#)

Methods

async

addParticipants (participantIds, options) → Promise containing (Object with [AddParticipantsResult](#) properties or string)

Adds a list of participants by ID to the group

Parameters

Name	Type	Optional Description
participantIds	(string or Array of string)	
options	AddParticipnatsOptions	An object thay handles options for adding participants

Returns

Promise containing (Object with [AddParticipantsResult](#) properties or string)

Returns an object with the resulting data or an error message as a string

async

approveGroupMembershipRequests (options) → Promise containing Array of [MembershipRequestActionResult](#)

Approves membership requests if any

Parameter

Name	Type	Optional Description
options	MembershipRequestActionOptions	Options for performing a membership request action

Returns

Promise containing Array of [MembershipRequestActionResult](#)

Returns an array of requester IDs whose membership requests were approved and an error for each requester, if any occurred during the operation. If there are no requests, an empty array will be returned

async

archive ()

Archives this chat

Inherited from
[Chat#archive](#)

async

changeLabels (labelIds) → Promise containing void

Add or remove labels to this Chat

Parameter

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

labelIds			
----------	--	--	--

Inherited from
[Chat#changeLabels](#)

Returns

async

clearMessages () → Promise containing Boolean

Clears all messages from the chat

Inherited from
[Chat#clearMessages](#)

Returns

result

async

clearState ()

Stops typing or recording in chat immediately.

Inherited from
[Chat#clearState](#)

async

delete () → Promise containing Boolean

Deletes the chat

Inherited from

[Chat#delete](#)

Returns

result

async

deletePicture () → Promise containing boolean

Deletes the group's picture.

Returns

Promise containing boolean

Returns true if the picture was properly deleted. This can return false if the user does not have the necessary permissions.

async

demoteParticipants (participantIds) → Promise containing {status: number}

Demotes participants by IDs to regular users

Parameter

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

participantIds	Array of string	
----------------	-----------------	--

Returns

Promise containing {status: number}

Object with status code indicating if the operation was successful

async

fetchMessages (searchOptions) → Promise containing Array of [Message](#)

Loads chat messages, sorted from earliest to latest.

Parameters

Name	Type	Optional	Description
Options for searching messages. Right now only limit and fromMe is supported.			
Values in searchOptions have the following properties:			
Name	Type	Optional	Description
searchOptions	limit	Yes	The amount of messages to return. If no limit is specified, the available messages will be returned. Note that the actual number of returned messages may be smaller if there aren't enough messages in the conversation. Set this to Infinity to load all messages.
	fromMe	Yes	Return only messages from the bot number or vise versa. To get all messages, leave the option undefined.

Inherited from [Chat#fetchMessages](#)
Returns

async

getContact () → Promise containing [Contact](#)

Returns the Contact that corresponds to this Chat.

Inherited from

[Chat#getContact](#)

Returns

async

getGroupMembershipRequests () → Promise containing Array of [GroupMembershipRequest](#)

Gets an array of membership requests

Returns

Promise containing Array of [GroupMembershipRequest](#)

An array of membership requests

async

getInviteCode () → Promise containing string

Gets the invite code for a specific group

Returns

Promise containing string

Group's invite code

async

getLabels () → Promise containing Array of [Label](#)

Returns array of all Labels assigned to this Chat

Inherited from

[Chat#getLabels](#)

Returns

async

leave () → Promise

Makes the bot leave the group

Returns

Promise

async

markUnread ()

Mark this chat as unread

Inherited from

[Chat#markUnread](#)

async

mute (unmuteDate)

Mutes this chat forever, unless a date is specified

Parameter

Name	Type	Optional	Description
unmuteDate			Date at which the Chat will be unmuted, leave as is to mute forever Value can be null.

Inherited from

[Chat#mute](#)

async

pin () → Promise containing boolean

Pins this chat

Inherited from

[Chat#pin](#)

Returns

New pin state. Could be false if the max number of pinned chats was reached.

async

promoteParticipants (participantIds) → Promise containing {status: number}

Promotes participants by IDs to admins

Parameter

Name	Type	Optional	Description
participantIds	Array of string		

Returns

Promise containing {status: number}

Object with status code indicating if the operation was successful

async

rejectGroupMembershipRequests (options) → Promise containing Array of [MembershipRequestActionResult](#)

Rejects membership requests if any

Parameter

Name	Type	Optional Description
options	MembershipRequestActionOptions	Options for performing a membership request action

Returns

Promise containing Array of [MembershipRequestActionResult](#)

Returns an array of requester IDs whose membership requests were rejected and an error for each requester, if any occurred during the operation. If there are no requests, an empty array will be returned

async

removeParticipants (participantIds) → Promise containing {status: number}

Removes a list of participants by ID to the group

Parameter

Name	Type	Optional Description
participantIds	Array of string	

Returns

Promise containing {status: number}

async

revokeInvite () → Promise containing string

Invalidates the current group invite code and generates a new one

Returns

Promise containing string

New invite code

async

sendMessage (content[, options]) → Promise containing [Message](#)

Send a message to this chat

Parameters

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

content			
---------	--	--	--

options	Yes		
---------	-----	--	--

Inherited from

[Chat#sendMessage](#)

Returns

Message that was just sent

async

sendSeen () → Promise containing Boolean

Set the message as seen

Inherited from

[Chat#sendSeen](#)

Returns

result

async

sendStateRecording ()

Simulate recording audio in chat. This will last for 25 seconds.

Inherited from

[Chat#sendStateRecording](#)

async

sendStateTyping ()

Simulate typing in chat. This will last for 25 seconds.

Inherited from

[Chat#sendStateTyping](#)

async

setDescription (description) → Promise containing boolean

Updates the group description

Parameter

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

description	string		
-------------	--------	--	--

Returns

Promise containing boolean

Returns true if the description was properly updated. This can return false if the user does not have the necessary permissions.

async

setInfoAdminsOnly ([adminsOnly]) → Promise containing boolean

Updates the group settings to only allow admins to edit group info (title, description, photo).

Parameter

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

adminsOnly	boolean	Yes	Enable or disable this option Defaults to true .
------------	---------	-----	---

Returns

Promise containing boolean

Returns true if the setting was properly updated. This can return false if the user does not have the necessary permissions.

async

setMessagesAdminsOnly ([adminsOnly]) → Promise containing boolean

Updates the group settings to only allow admins to send messages.

Parameter

Name	Type	Optional	Description
adminsOnly	boolean	Yes	Enable or disable this option Defaults to true .

Returns

Promise containing boolean

Returns true if the setting was properly updated. This can return false if the user does not have the necessary permissions.

async

setPicture (media) → Promise containing boolean

Sets the group's picture.

Parameter

Name	Type	Optional	Description
media	MessageMedia		

Returns

Promise containing boolean

Returns true if the picture was properly updated. This can return false if the user does not have the necessary permissions.

async

setSubject (subject) → Promise containing boolean

Updates the group subject

Parameter

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

subject	string		
---------	--------	--	--

Returns

Promise containing boolean

Returns true if the subject was properly updated. This can return false if the user does not have the necessary permissions.

async

unarchive ()

un-archives this chat

Inherited from

[Chat#unarchive](#)

async

unmute ()

Unmutes this chat

Inherited from

[Chat#unmute](#)

async

unpin () → Promise containing boolean

Unpins this chat

Inherited from

[Chat#unpin](#)

Returns

New pin state

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.

[whatsapp-web.js](#) 1.23.0
class

GroupNotification

Source: [structures/ GroupNotification. js:9](#)

Represents a GroupNotification on WhatsApp

Properties

[author](#)
[body](#)
[chatId](#)

[id](#)
[recipientIds](#)
[timestamp](#)

[type](#)

Methods

[getChat\(\)](#)
[getContact\(\)](#)

[getRecipients\(\)](#)
[reply\(content, options\)](#)

new GroupNotification ()

Extends
[Base](#)

Properties

author string

ContactId for the user that produced the GroupNotification.

body string

Extra content

chatId string

ID for the Chat that this groupNotification was sent for.

id object

ID that represents the groupNotification

recipientIds Array of string

Contact IDs for the users that were affected by this GroupNotification.

timestamp number

Unix timestamp for when the groupNotification was created

type [GroupNotificationTypes](#)

GroupNotification type

Methods

getChat () → Promise containing [Chat](#)

Returns the Chat this groupNotification was sent in

Returns

Promise containing [Chat](#)

getContact () → Promise containing [Contact](#)

Returns the Contact this GroupNotification was produced by

Returns

Promise containing [Contact](#)

async

getRecipients () → Promise containing Array of [Contact](#)

Returns the Contacts affected by this GroupNotification.

Returns

Promise containing Array of [Contact](#)

async

reply (content, options) → Promise containing [Message](#)

Sends a message to the same chat this GroupNotification was produced in.

Parameters

Name	Type	Optional	Description
content	(string, MessageMedia , or Location)		
options	object		

Returns

Promise containing [Message](#)

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.

[whatsapp-web.js 1.23.0](#)
class

InterfaceController

Source: [util/InterfaceController.js:6](#)

Interface Controller

Methods

[checkFeatureStatus\(feature\)](#)
[closeRightDrawer\(\)](#)
[disableFeatures\(features\)](#)
[enableFeatures\(features\)](#)

[getFeatures\(\)](#)
[openChatDrawer\(chatId\)](#)
[openChatSearch\(chatId\)](#)
[openChatWindow\(chatId\)](#)

[openChatWindowAt\(msgId\)](#)
[openMessageDrawer\(msgId\)](#)

new InterfaceController ()

Methods

async

checkFeatureStatus (feature)

Check if Feature is enabled

Parameter

Name	Type	Optional	Description
feature	string		status to check

async

closeRightDrawer ()

Closes the Right Drawer

async

disableFeatures (features)

Disable Features

Parameter

Name	Type	Optional	Description
features	Array of string		to be disabled

async

enableFeatures (features)

Enable Features

Parameter

Name	Type	Optional	Description
features	Array of string		to be enabled

async

getFeatures ()

Get all Features

async

openChatDrawer (chatId)

Opens the Chat Drawer

Parameter

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

chatId	string		ID of the chat drawer that will be opened
--------	--------	--	---

async

openChatSearch (chatId)

Opens the Chat Search

Parameter

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

chatId	string		ID of the chat search that will be opened
--------	--------	--	---

async

openChatWindow (chatId)

Opens the Chat Window

Parameter

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

chatId	string		ID of the chat window that will be opened
--------	--------	--	---

async

openChatWindowAt (msgId)

Opens or Scrolls the Chat Window to the position of the message

Parameter

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

msgId	string		ID of the message that will be scrolled to
-------	--------	--	--

async

openMessageDrawer (msgId)

Opens the Message Drawer

Parameter

Name	Type	Optional	Description
msgId	string		ID of the message drawer that will be opened

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.

[whatsapp-web.js 1.23.0](#)
class

Label

Source: [structures/ Label.js:10](#)

WhatsApp Business Label information

Properties

[hexColor](#)

[id](#)

[name](#)

Method

[getChats\(\)](#)

new Label (client, labelData)

Parameters

Name	Type	Optional	Description
client			
labelData			

Properties

hexColor string

Label hex color

id string

Label ID

name string

Label name

Method

async

getChats () → Promise containing Array of [Chat](#)

Get all chats that have been assigned this Label

Returns

Promise containing Array of [Chat](#)

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.

[whatsapp-web.js 1.23.0](#)
class

LegacySessionAuth

Source: [authStrategies/ LegacySessionAuth.js:16](#)

Legacy session auth strategy Not compatible with multi-device accounts.

new LegacySessionAuth (options)

Parameters

Name	Type	Optional	Description
			options
options			Values in options have the following properties:

Name	Type	Optional	Description	Name	Type	Optional	Description
				restartOnAuthFail			Restart client with a new session (i.e. use null 'session' var) if authentication fails
				session			Whatsapp session to restore. If not set, will start a new session
				session.WABrowserId			
				session.WASecretBundle			
				session.WAToken1			
				session.WAToken2			

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.

[whatsapp-web.js 1.23.0](#)
class

List

Source: [structures/List.js:8](#)

Message type List

Properties

[buttonText](#)
[description](#)

[footer](#)
[sections](#)

[title](#)

Method

[_format\(sections\)](#)

**new List (body, buttonText, sections,
title, footer)**

Parameters

Name	Type	Optional	Description
body			
buttonText			
sections			
title			Value can be null.
footer			Value can be null.

Properties

buttonText string

List button text

description string

Message body

footer string

footer of message

sections Array of any

sections of message

title string

title of message

Method

`_format (sections) → Array of any`

Creates section array from simple array

Example

Input: `[{title: 'sectionTitle', rows: [{id: 'customId', title: 'ListItem2', description: 'desc`
Returns: `[{'title': 'sectionTitle', 'rows': [{'rowId': 'customId', 'title': 'ListItem1', 'descri`

Parameter

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

sections	Array of any		
----------	--------------	--	--

Returns

Array of any

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.

[whatsapp-web.js 1.23.0](#)
class

LocalAuth

Source: [authStrategies/ LocalAuth.js:13](#)

Local directory-based authentication

new LocalAuth (options)

Parameters

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

options			
---------	--	--	--

Values in options have the following properties:

options			
---------	--	--	--

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

clientId			Client id to distinguish instances if you are using multiple, otherwise
----------	--	--	---

Name	Type	Optional	Description
			keep null if you are using only one instance
		dataPath	Change the default path for saving session files, default is: <code>"./.wwebjs_auth/"</code>

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.

[whatsapp-web.js 1.23.0](#)
class

LocalWebCache

Source: [webCache/ LocalWebCache.js:12](#)

LocalWebCache - Fetches a WhatsApp Web version from a local file store

new LocalWebCache (options)

Parameters

Name	Type	Optional	Description								
		options	Values in options have the following properties:								
			<table> <tr> <th>Name</th><th>Type</th><th>Optional</th><th>Description</th></tr> <tr> <td>options</td><td></td><td>path</td><td>Path to the directory where cached versions are saved, default is: <code>"./.wwebjs_cache/"</code></td></tr> </table>	Name	Type	Optional	Description	options		path	Path to the directory where cached versions are saved, default is: <code>"./.wwebjs_cache/"</code>
Name	Type	Optional	Description								
options		path	Path to the directory where cached versions are saved, default is: <code>"./.wwebjs_cache/"</code>								

Properties

address (string or undefined)

Location address

description (string or undefined)

Location full description

latitude number

Location latitude

longitude number

Location longitude

name (string or undefined)

Name for the location

url (string or undefined)

URL address to be shown within a location message

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.

[whatsapp-web.js 1.23.0](#)
class

Message

Source: [structures/ Message.js:16](#)

Represents a Message on WhatsApp

Properties

[ack](#)
[author](#)
[body](#)
[broadcast](#)
[deviceType](#)
[duration](#)
[forwardingScore](#)
[from](#)
[fromMe](#)
[hasMedia](#)

[hasQuotedMsg](#)
[hasReaction](#)
[id](#)
[inviteV4](#)
[isEphemeral](#)
[isForwarded](#)
[isGif](#)
[isStarred](#)
[isStatus](#)
[links](#)

[location](#)
[mediaKey](#)
[mentionedIds](#)
[orderId](#)
[rawData](#)
[timestamp](#)
[to](#)
[token](#)
[type](#)
[vCards](#)

Methods

[acceptGroupV4Invite\(\)](#)
[delete\(everyone\)](#)
[downloadMedia\(\)](#)
[edit\(content\[, options\]\)](#)
[forward\(chat\)](#)
[getChat\(\)](#)

[getContact\(\)](#)
[getInfo\(\)](#)
[getMentions\(\)](#)
[getOrder\(\)](#)
[getPayment\(\)](#)
[getQuotedMessage\(\)](#)

[getReactions\(\)](#)
[react\(reaction\)](#)
[reload\(\)](#)
[reply\(content\[, chatId\]\[, options\]\)](#)
[star\(\)](#)
[unstar\(\)](#)

new Message ()

Extends

[Base](#)

Properties

ack [MessageAck](#)

ACK status for the message

author string

If the message was sent to a group, this field will contain the user that sent the message.

body string

Message content

broadcast boolean

Indicates if the message was a broadcast

deviceType string

String that represents from which device type the message was sent

duration string

Indicates the duration of the message in seconds

forwardingScore number

Indicates how many times the message was forwarded.

The maximum value is 127.

from string

ID for the Chat that this message was sent to, except if the message was sent by the current user.

fromMe boolean

Indicates if the message was sent by the current user

hasMedia boolean

Indicates if the message has media available for download

hasQuotedMsg boolean

Indicates if the message was sent as a reply to another message.

hasReaction boolean

Indicates whether there are reactions to the message

id object

ID that represents the message

inviteV4 object

Group Invite Data

isEphemeral boolean

Indicates if the message will disappear after it expires

isForwarded boolean

Indicates if the message was forwarded

isGif boolean

Indicates whether the message is a Gif

isStarred boolean

Indicates if the message was starred

isStatus boolean

Indicates if the message is a status update

links Array of {link: string, isSuspicious: boolean}

Links included in the message.

location [Location](#)

Location information contained in the message, if the message is type "location"

mediaKey string

MediaKey that represents the sticker 'ID'

mentionedIds Array of string

Indicates the mentions in the message body.

orderId string

Order ID for message type ORDER

rawData object

Returns message in a raw format

timestamp number

Unix timestamp for when the message was created

to string

ID for who this message is for.

If the message is sent by the current user, it will be the Chat to which the message is being sent. If the message is sent by another user, it will be the ID for the current user.

token string

Order Token for message type ORDER

type [MessageTypes](#)

Message type

vCards Array of string

List of vCards contained in the message.

Methods

async

acceptGroupV4Invite () → Promise containing Object

Accept Group V4 Invite

Returns

Promise containing Object

async

delete (everyone)

Deletes a message from the chat

Parameter

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

everyone	boolean		If true and the message is sent by the current user or the user is an admin, will delete it for everyone in the chat. Value can be null.
----------	---------	--	---

async

downloadMedia () → Promise containing [MessageMedia](#)

Downloads and returns the attached message media

Returns

Promise containing [MessageMedia](#)

async

edit (content[, options]) → Promise containing nullable [Message](#)

Edits the current message.

Parameters

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

content	string		
---------	--------	--	--

options	MessageEditOptions	Yes	Options used when editing the message
---------	--------------------	-----	---------------------------------------

Returns

Promise containing nullable [Message](#)

async

forward (chat) → Promise

Forwards this message to another chat (that you chatted before, otherwise it will fail)

Parameter

Name	Type	Optional	Description
chat	(string or Chat)		Chat model or chat ID to which the message will be forwarded

Returns

Promise

getChat () → Promise containing [Chat](#)

Returns the Chat this message was sent in

Returns

Promise containing [Chat](#)

getContact () → Promise containing [Contact](#)

Returns the Contact this message was sent from

Returns

Promise containing [Contact](#)

async

getInfo () → Promise containing nullable [MessageInfo](#)

Get information about message delivery status. May return null if the message does not exist or is not sent by you.

Returns

Promise containing nullable [MessageInfo](#)

async

getMentions () → Promise containing Array of [Contact](#)

Returns the Contacts mentioned in this message

Returns

Promise containing Array of [Contact](#)

async

getOrder () → Promise containing [Order](#)

Gets the order associated with a given message

Returns

Promise containing [Order](#)

async

getPayment () → Promise containing [Payment](#)

Gets the payment details associated with a given message

Returns

Promise containing [Payment](#)

async

getQuotedMessage () → Promise containing [Message](#)

Returns the quoted message, if any

Returns

Promise containing [Message](#)

async

getReactions () → Promise containing Array of [ReactionList](#)

Gets the reactions associated with the given message

Returns

Promise containing Array of [ReactionList](#)

async

react (reaction) → Promise

React to this message with an emoji

Parameter

Name	Type	Optional	Description
reaction	string		Emoji to react with. Send an empty string to remove the reaction.

Returns

Promise

async

reload () → Promise containing [Message](#)

Reloads this Message object's data in-place with the latest values from WhatsApp Web. Note that the Message must still be in the web app cache for this to work, otherwise will return null.

Returns

Promise containing [Message](#)

async

reply (content[, chatId][, options]) → Promise containing [Message](#)

Sends a message as a reply to this message. If chatId is specified, it will be sent through the specified Chat. If not, it will send the message in the same Chat as the original message was sent.

Parameters

Name	Type	Optional	Description
content	(string, MessageMedia , or Location)		
chatId	string	Yes	
options	MessageSendOptions	Yes	

Returns

Promise containing [Message](#)

async

star ()

Stars this message

async

unstar ()

Unstars this message

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.

[whatsapp-web.js 1.23.0](#)
class

MessageMedia

Source: [structures/ MessageMedia.js:16](#)

Media attached to a message

Properties

[data](#)
[filename](#)

[filesize](#)
[mimetype](#)

Methods

[fromFilePath\(filePath\)](#)
[fromUrl\(url\[, options\]\)](#)

new MessageMedia (mimetype, data, filename, filesize)

Parameters

Name	Type	Optional	Description
mimetype			MIME type of the attachment
data			Base64-encoded data of the file
filename			Document file name. Value can be null Value can be null.
filesize			Document file size in bytes. Value can be null

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

Value can be null.

Properties

data string

Base64 encoded data that represents the file

filename nullable string

Document file name. Value can be null

filesize nullable number

Document file size in bytes. Value can be null

mimetype string

MIME type of the attachment

Methods

static

fromFilePath (filePath) → [MessageMedia](#)

Creates a MessageMedia instance from a local file path

Parameter

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

filePath string

Returns

[MessageMedia](#)

async static

fromUrl (url[, options]) → Promise containing [MessageMedia](#)

Creates a MessageMedia instance from a URL

Parameters

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

url	string		
-----	--------	--	--

Values in options have the following properties:

	Name	Type	Optional	Description
	unsafeMime	boolean	Yes	Defaults to false .
options Object	filename	string	Yes	
	client	object	Yes	
	reqOptions	object	Yes	
	reqOptions.size	number	Yes	Defaults to 0 .

Returns

Promise containing [MessageMedia](#)

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.

[whatsapp-web.js 1.23.0](#)
class

NoAuth

Source: [authStrategies/ NoAuth.js:9](#)

No session restoring functionality Will need to authenticate via QR code every time

new NoAuth ()

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.

[whatsapp-web.js 1.23.0](#)
class

Order

Source: [structures/ Order. js:10](#)

Represents a Order on WhatsApp

Properties

[createdAt](#)
[currency](#)

[subtotal](#)
[total](#)

new Order ()

Extends
[Base](#)

Properties

createdAt number

Order Created At

currency string

Order Currency

subtotal string

Order Subtotal

total string

Order Total

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.

[whatsapp-web. js 1. 23. 0](#)
class

Poll

Source: [structures/ Poll. js:11](#)

Represents a Poll on WhatsApp

Properties

[options](#)

[pollName](#)

[pollOptions](#)

new Poll (pollName, pollOptions, options)

Parameters

Name	Type	Optional	Description
pollName			
pollOptions			
options			

Properties

options [PollSendOptions](#)

The send options for the poll

pollName string

The name of the poll

pollOptions Array of {name: string, localId: number}

The array of poll options

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.

[whatsapp-web.js 1.23.0](#)
class

PrivateChat

Source: [structures/PrivateChat.js:9](#)

Represents a Private Chat on WhatsApp

Properties

[archived](#)

[id](#)

[isGroup](#)

[isMuted](#)

[isReadOnly](#)

[lastMessage](#)

[muteExpiration](#)

[name](#)

[pinned](#)

[timestamp](#)

[unreadCount](#)

Methods

[archive\(\)](#)

[changeLabels\(labelIds\)](#)

[clearMessages\(\)](#)

[clearState\(\)](#)

[delete\(\)](#)

[fetchMessages\(searchOptions\)](#)

[getContact\(\)](#)

[getLabels\(\)](#)

[markUnread\(\)](#)

[mute\(unmuteDate\)](#)

[pin\(\)](#)

[sendMessage\(content\[, options\]\)](#)

[sendSeen\(\)](#)

[sendStateRecording\(\)](#)

[sendStateTyping\(\)](#)

[unarchive\(\)](#)

[unmute\(\)](#)

[unpin\(\)](#)

new PrivateChat ()

Extends

[Chat](#)

Properties

archived unknown

Indicates if the Chat is archived

Inherited from

[Chat#archived](#)

id unknown

ID that represents the chat

Inherited from

[Chat#id](#)

isGroup unknown

Indicates if the Chat is a Group Chat

Inherited from

[Chat#isGroup](#)

isMuted unknown

Indicates if the chat is muted or not

Inherited from

[Chat#isMuted](#)

isReadOnly unknown

Indicates if the Chat is readonly

Inherited from

[Chat#isReadOnly](#)

lastMessage unknown

Last message fo chat

Inherited from

[Chat#lastMessage](#)

muteExpiration unknown

Unix timestamp for when the mute expires

Inherited from

[Chat#muteExpiration](#)

name unknown

Title of the chat

Inherited from

[Chat#name](#)

pinned unknown

Indicates if the Chat is pinned

Inherited from
[Chat#pinned](#)

timestamp unknown

Unix timestamp for when the last activity occurred

Inherited from
[Chat#timestamp](#)

unreadCount unknown

Amount of messages unread

Inherited from
[Chat#unreadCount](#)

Methods

async

archive ()

Archives this chat

Inherited from
[Chat#archive](#)

async

changeLabels (labelIds) → Promise containing void

Add or remove labels to this Chat

Parameter

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

labelIds			
----------	--	--	--

Inherited from
[Chat#changeLabels](#)

Returns

async

clearMessages () → Promise containing Boolean

Clears all messages from the chat

Inherited from

[Chat#clearMessages](#)

Returns

result

async

clearState ()

Stops typing or recording in chat immediately.

Inherited from

[Chat#clearState](#)

async

delete () → Promise containing Boolean

Deletes the chat

Inherited from

[Chat#delete](#)

Returns

result

async

fetchMessages (searchOptions) → Promise containing Array of [Message](#)

Loads chat messages, sorted from earliest to latest.

Parameters

Name	Type	Optional	Description
			Options for searching messages. Right now only limit and fromMe is supported.
searchOptions			Values in searchOptions have the following properties:

Name	Type	Optional	Description
Name	Type	Optional	Description
			The amount of messages to return. If no limit is specified, the available messages will be returned. Note that the actual number of returned messages may be smaller if there aren't enough messages in the conversation. Set this to Infinity to load all messages.
limit		Yes	
			Return only messages from the bot number or vice versa. To get all messages, leave the option undefined.
fromMe		Yes	

Inherited from
[Chat#fetchMessages](#)
Returns

async

getContact () → Promise containing [Contact](#)

Returns the Contact that corresponds to this Chat.

Inherited from
[Chat#getContact](#)
Returns

async

getLabels () → Promise containing Array of [Label](#)

Returns array of all Labels assigned to this Chat

Inherited from
[Chat#getLabels](#)

Returns

async

markUnread ()

Mark this chat as unread

Inherited from

[Chat#markUnread](#)

async

mute (unmuteDate)

Mutes this chat forever, unless a date is specified

Parameter

Name	Type	Optional	Description
unmuteDate	Date		Date at which the Chat will be unmuted, leave as is to mute forever
			Value can be null.

Inherited from

[Chat#mute](#)

async

pin () → Promise containing boolean

Pins this chat

Inherited from

[Chat#pin](#)

Returns

New pin state. Could be false if the max number of pinned chats was reached.

async

sendMessage (content[, options]) → Promise containing [Message](#)

Send a message to this chat

Parameters

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

content			
---------	--	--	--

options	Yes		
---------	-----	--	--

Inherited from

[Chat#sendMessage](#)

Returns

Message that was just sent

async

sendSeen () → Promise containing Boolean

Set the message as seen

Inherited from

[Chat#sendSeen](#)

Returns

result

async

sendStateRecording ()

Simulate recording audio in chat. This will last for 25 seconds.

Inherited from

[Chat#sendStateRecording](#)

async

sendStateTyping ()

Simulate typing in chat. This will last for 25 seconds.

Inherited from

[Chat#sendStateTyping](#)

async

unarchive ()

un-archives this chat

Inherited from

[Chat#unarchive](#)

async

unmute ()

Unmutes this chat

Inherited from

[Chat#unmute](#)

async

unpin () → Promise containing boolean

Unpins this chat

Inherited from

[Chat#unpin](#)

Returns

New pin state

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.

[whatsapp-web.js 1.23.0](#)
class

PrivateContact

Source: [structures/PrivateContact.js:9](#)

Represents a Private Contact on WhatsApp

Properties

[id](#)

[isBlocked](#)

[isBusiness](#)

[isEnterprise](#)

[isGroup](#)

[isMe](#)

[isMyContact](#)

[isUser](#)

[isWABusiness](#)

[name](#)

[number](#)

[pushname](#)

[shortName](#)

Methods

[block\(\)](#)

[getAbout\(\)](#)

[getChat\(\)](#)

[getCommonGroups\(\)](#)

[getCountryCode\(\)](#)

[getFormattedNumber\(\)](#)

[getProfilePicUrl\(\)](#)

[unblock\(\)](#)

new PrivateContact ()

Extends

[Contact](#)

Properties

id unknown

ID that represents the contact

Inherited from

[Contact#id](#)

isBlocked unknown

Indicates if you have blocked this contact

Inherited from

[Contact#isBlocked](#)

isBusiness unknown

Indicates if the contact is a business contact

Inherited from

[Contact#isBusiness](#)

isEnterprise unknown

Indicates if the contact is an enterprise contact

Inherited from

[Contact#isEnterprise](#)

isGroup unknown

Indicates if the contact is a group contact

Inherited from
[Contact#isGroup](#)

isMe unknown

Indicates if the contact is the current user's contact

Inherited from
[Contact#isMe](#)

isMyContact unknown

Indicates if the number is saved in the current phone's contacts

Inherited from
[Contact#isMyContact](#)

isUser unknown

Indicates if the contact is a user contact

Inherited from
[Contact#isUser](#)

isWAContact unknown

Indicates if the number is registered on WhatsApp

Inherited from
[Contact#isWAContact](#)

name unknown

The contact's name, as saved by the current user

Inherited from
[Contact#name](#)

number unknown

Contact's phone number

Inherited from
[Contact#number](#)

pushname unknown

The name that the contact has configured to be shown publically

Inherited from
[Contact#pushname](#)

shortName unknown

A shortened version of name

Inherited from

[Contact#shortName](#)

Methods

async

block () → Promise containing boolean

Blocks this contact from WhatsApp

Inherited from

[Contact#block](#)

Returns

async

getAbout () → Promise containing nullable string

Gets the Contact's current "about" info. Returns null if you don't have permission to read their status.

Inherited from

[Contact#getAbout](#)

Returns

async

getChat () → Promise containing [Chat](#)

Returns the Chat that corresponds to this Contact. Will return null when getting chat for currently logged in user.

Inherited from

[Contact#getChat](#)

Returns

async

getCommonGroups () → Promise containing Array of WAWebJS.ChatId

Gets the Contact's common groups with you. Returns empty array if you don't have any common group.

Inherited from

[Contact#getCommonGroups](#)

Returns

async

getCountryCode () → Promise containing string

Returns the contact's countrycode, ([\[email protected\]](#)) => (1)

Inherited from

[Contact#getCountryCode](#)

Returns

async

getFormattedNumber () → Promise containing string

Returns the contact's formatted phone number, ([\[email protected\]](#)) => (+1 (234) 5678-901)

Inherited from

[Contact#getFormattedNumber](#)

Returns

async

getProfilePicUrl () → Promise containing string

Returns the contact's profile picture URL, if privacy settings allow it

Inherited from

[Contact#getProfilePicUrl](#)

Returns

async

unblock () → Promise containing boolean

Unblocks this contact from WhatsApp

Inherited from

[Contact#unblock](#)

Returns

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.

[whatsapp-web. js 1. 23. 0](#)

class

Product

Source: [structures/ Product. js:10](#)

Represents a Product on WhatsAppBusiness

Properties

[currency](#)
[data](#)
[id](#)

[name](#)
[price](#)
[quantity](#)

[thumbnailUrl](#)

new Product ()

Extends
[Base](#)

Properties

currency string

Currency

data

Product metadata

id string

Product ID

name string

Product Name

price string

Price

quantity number

Product Quantity

thumbnailUrl string

Product Thumbnail

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.

[whatsapp-web.js 1.23.0](#)

class

Reaction

Source: [structures/ Reaction. js:9](#)

Represents a Reaction on WhatsApp

Properties

[ack](#)

[id](#)

[msgId](#)

[orphan](#)

[orphanReason](#)

[reaction](#)

[read](#)

[senderId](#)

[timestamp](#)

new Reaction ()

Extends

[Base](#)

Properties

ack nullable number

ACK

id object

Reaction ID

msgId object

Message ID

orphan number

Orphan

orphanReason nullable string

Orphan reason

reaction string

Reaction

read boolean

Read

senderId string

Sender ID

timestamp number

Unix timestamp for when the reaction was created

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.

[whatsapp-web. js 1. 23. 0](#)
class

RemoteAuth

Source: [authStrategies/ RemoteAuth. js:26](#)

Remote-based authentication

new RemoteAuth (options)

Parameters

Name	Type	Optional	Description
			options
			Values in options have the following properties:
	</		

Name	Type	Optional	Description
			using only one instance
		dataPath	Change the default path for saving session files, default is: "./.wwebjs_auth/"
		backupSyncIntervalMs	Sets the time interval for periodic session backups. Accepts values starting from 60000ms {1 minute}

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.

[whatsapp-web.js](#) 1.23.0
class

RemoteWebCache

Source: [webCache/ RemoteWebCache.js:10](#)

RemoteWebCache - Fetches a WhatsApp Web version index from a remote server

new RemoteWebCache (options)

Parameters

Name	Type	Optional	Description
		options	
		options	Values in options have the following properties:
		remotePath	Endpoint that should be used to fetch the

Name **Type** **Optional** **Description**

Name

Type **Optional** **Description**

version index. Use {version} as a placeholder for the version number.

strict

If true, will throw an error if the requested version can't be fetched. If false, will resolve to the latest version. Defaults to false.

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.

[whatsapp-web.js 1.23.0](#)
class

Util

Source: [util/ Util.js:14](#)

Utility methods

Methods

[formatImageToWebpSticker\(media\)](#)
[formatToWebpSticker\(media, metadata\)](#)

[formatVideoToWebpSticker\(media\)](#)
[setFfmpegPath\(path\)](#)

new Util ()

Methods

async static

formatImageToWebpSticker (media) → Promise containing [MessageMedia](#)

Formats a image to webp

Parameter

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

media	MessageMedia		
-------	------------------------------	--	--

Returns

Promise containing [MessageMedia](#)

media in webp format

async static

formatToWebpSticker (media, metadata) → Promise containing [MessageMedia](#)

Formats a media to webp

Parameters

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

media	MessageMedia		
-------	------------------------------	--	--

metadata	StickerMetadata		
----------	---------------------------------	--	--

Returns

Promise containing [MessageMedia](#)

media in webp format

async static

formatVideoToWebpSticker (media) → Promise containing [MessageMedia](#)

Formats a video to webp

Parameter

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

media	MessageMedia		
-------	------------------------------	--	--

Returns

Promise containing [MessageMedia](#)

media in webp format

static

setFfmpegPath (path)

Configure ffmpeg path

Parameter

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

path	string		
------	--------	--	--

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.

[whatsapp-web.js 1.23.0](#)
class

WebCache

Source: [webCache/ WebCache.js:4](#)

Default implementation of a web version cache that does nothing.

new WebCache ()

Generated by [JSDoc](#) 3.6.11 on October 21, 2023.