INIZIO

Get Information

GET
/
Try it
Path Parameters

instance
string
required
ID of the instance to connect
Response
200 - application/json
Ok

status
integer
The HTTP status of the response

message
string
Descriptive message about the current state of the API

version
string
The current version of the API

swagger
string
URL to the API's Swagger documentation

manager
string
URL to the API manager

documentation
string
URL to the detailed API documentation

Get information about your EvolutionAPI Curl
```
curl --request GET \
  --url https://{server-url}/
```

Get information about your EvolutionAPI javascript

```
const url = 'https://{server-url}/';
const options = {method: 'GET', body: undefined};

try {
  const response = await fetch(url, options);
  const data = await response.json();
  console.log(data);
} catch (error) {
  console.error(error);
}
```

200

```
{
  "status": 200,
  "message": "Welcome to the Evolution API, it is working!",
  "version": "1.7.4",
  "swagger": "http://example.evolution-api.com/docs",
  "manager": "http://example.evolution-api.com/manager",
  "documentation": "https://doc.evolution-api.com"
}
```


ISTANZA

Create Instance
POST
/
instance
/
create
Try it
Authorizations

apikey

string
header
required
Your authorization key header
Body
application/json

instanceName
string
required
instance (Instance name)

integration
enum<string>
required
WhatsApp engine
Available options:
WHATSAPP-BAILEYS,
WHATSAPP-BUSINESS

token
string
apikey (Enter or leave empty to create dynamically)

qrcode
boolean
Create QR Code automatically after creation

number
string
559999999999 (Instance owner number with Country Code)

rejectCall
boolean
Reject WhatsApp calls automatically

msgCall
string
Message to be sent when a call is rejected automatically

groupsIgnore
boolean
Ignore group messages

alwaysOnline
boolean
Keep WhatsApp always online

readMessages
boolean
Send read receipts to received messages

readStatus
boolean
Show sent messages read status

syncFullHistory
boolean
Syncronize full WhatsApp history with EvolutionAPI

proxyHost
string
proxy host

proxyPort
string
proxy port

proxyProtocol
string
proxy protocol

proxyUsername
string
proxy Username

proxyPassword
string
proxy password

webhook
object
Webhook URL
Show child attributes

rabbitmq
object
Show child attributes

sqs
object
Enable SQS
Show child attributes

chatwootAccountId
integer
Chatwoot account ID

chatwootToken
string
Chatwoot authentication token

chatwootUrl
string
Chatwoot server URL

chatwootSignMsg
boolean
Send message signature on Chatwoot

chatwootReopenConversation
boolean
Reopen conversation on Chatwoot

chatwootConversationPending
boolean
TODO

chatwootImportContacts
boolean
Import Chatwoot contacts

chatwootNameInbox
string
Name inbox chatwoot

chatwootMergeBrazilContacts
boolean
TODO

chatwootImportMessages
boolean
Import chatwoot messages

chatwootDaysLimitImportMessages
integer
Limit message import chatwoot

chatwootOrganization
string
Evolution Bot

chatwootLogo
string
https://evolution-api.com/files/evolution-api-favicon.png
Response
201
application/json
Created

instance
object
Show child attributes

hash
object
Show child attributes

settings
object
Show child attributes

Create Instance curl

```
curl --request POST \
  --url https://{server-url}/instance/create \
  --header 'Content-Type: application/json' \
  --header 'apikey: <api-key>' \
  --data '{
  "instanceName": "<string>",
  "token": "<string>",
  "qrcode": true,
  "number": "<string>",
  "integration": "WHATSAPP-BAILEYS",
  "rejectCall": true,
  "msgCall": "<string>",
  "groupsIgnore": true,
  "alwaysOnline": true,
  "readMessages": true,
  "readStatus": true,
  "syncFullHistory": true,
  "proxyHost": "<string>",
  "proxyPort": "<string>",
  "proxyProtocol": "<string>",
  "proxyUsername": "<string>",
  "proxyPassword": "<string>",
  "webhook": {
    "url": "<string>",
    "byEvents": true,
    "base64": true,
    "headers": {
```

```
    "authorization": "<string>",
    "Content-Type": "<string>"
   },
   "events": [
    "APPLICATION_STARTUP"
   ]
  },
  "rabbitmq": {
   "enabled": true,
   "events": [
    "APPLICATION_STARTUP"
   ]
  },
  "sqs": {
   "enabled": true,
   "events": [
    "APPLICATION_STARTUP"
   ]
  },
  "chatwootAccountId": 123,
  "chatwootToken": "<string>",
  "chatwootUrl": "<string>",
  "chatwootSignMsg": true,
  "chatwootReopenConversation": true,
  "chatwootConversationPending": true,
  "chatwootImportContacts": true,
  "chatwootNameInbox": "<string>",
  "chatwootMergeBrazilContacts": true,
  "chatwootImportMessages": true,
  "chatwootDaysLimitImportMessages": 123,
  "chatwootOrganization": "<string>",
  "chatwootLogo": "<string>"
}'
```

Create Instance javascript

```
const url = 'https://{server-url}/instance/create';
const options = {
  method: 'POST',
  headers: {apikey: '<api-key>', 'Content-Type': 'application/json'},
  body:
'{"instanceName":"<string>","token":"<string>","qrcode":true,"number":"<string>","integration":"
WHATSAPP-
BAILEYS","rejectCall":true,"msgCall":"<string>","groupsIgnore":true,"alwaysOnline":true,"readMes
sages":true,"readStatus":true,"syncFullHistory":true,"proxyHost":"<string>","proxyPort":"<string>"
,"proxyProtocol":"<string>","proxyUsername":"<string>","proxyPassword":"<string>","webhook":{
"url":"<string>","byEvents":true,"base64":true,"headers":{"authorization":"<string>","Content-
```

Type":"<string>"},"events":["APPLICATION_STARTUP"]},"rabbitmq":{"enabled":true,"events":["APPLICATION_STARTUP"]},"sqs":{"enabled":true,"events":["APPLICATION_STARTUP"]},"chatwootAccountId":123,"chatwootToken":"<string>","chatwootUrl":"<string>","chatwootSignMsg":true,"chatwootReopenConversation":true,"chatwootConversationPending":true,"chatwootImportContacts":true,"chatwootNameInbox":"<string>","chatwootMergeBrazilContacts":true,"chatwootImportMessages":true,"chatwootDaysLimitImportMessages":123,"chatwootOrganization":"<string>","chatwootLogo":"<string>"}'
};

```
try {
  const response = await fetch(url, options);
  const data = await response.json();
  console.log(data);
} catch (error) {
  console.error(error);
}
```

201


```
 {
  "instance": {
    "instanceName": "teste-docs",
    "instanceId": "af6c5b7c-ee27-4f94-9ea8-192393746ddd",
    "webhook_wa_business": null,
    "access_token_wa_business": "",
    "status": "created"
  },
  "hash": {
    "apikey": "123456"
  },
  "settings": {
    "reject_call": false,
    "msg_call": "",
    "groups_ignore": true,
    "always_online": false,
    "read_messages": false,
    "read_status": false,
    "sync_full_history": false
  }
}
```


403


```
{
  "status": 403,
  "error": "Forbidden",
  "response": {
    "message": [
      "This name \"instance-example-name\" is already in use."
    ]
```

```
  }
}
```

Fetch Instances

GET
/
instance
/
fetchInstances
Try it
Authorizations

apikey
string
header
required
Your authorization key header
Query Parameters

instanceName
string
Name of the instance to be fetched

instanceId
string
ID of the instance to be fetched
Response
200 - application/json
Ok

status
integer
The HTTP status of the response

error
string
The error message indicating the type of error

response
object

Show child attributes

Fetch Instances curl

```
curl --request GET \
 --url https://{server-url}/instance/fetchInstances \
 --header 'apikey: <api-key>'
```

Fetch Instances javascript

```
const url = 'https://{server-url}/instance/fetchInstances';
const options = {method: 'GET', headers: {apikey: '<api-key>'}, body: undefined};

try {
 const response = await fetch(url, options);
 const data = await response.json();
 console.log(data);
} catch (error) {
 console.error(error);
}
```

200


```
[
 {
  "instance": {
   "instanceName": "example-name",
   "instanceId": "421a4121-a3d9-40cc-a8db-c3a1df353126",
   "owner": "553198296801@s.whatsapp.net",
   "profileName": "Guilherme Gomes",
   "profilePictureUrl": null,
   "profileStatus": "This is the profile status.",
   "status": "open",
   "serverUrl": "https://example.evolution-api.com",
   "apikey": "B3844804-481D-47A4-B69C-F14B4206EB56",
   "integration": {
    "integration": "WHATSAPP-BAILEYS",
    "webhook_wa_business": "https://example.evolution-
api.com/webhook/whatsapp/db5e11d3-ded5-4d91-b3fb-48272688f206"
   }
  }
 },
 {
  "instance": {
   "instanceName": "teste-docs",
   "instanceId": "af6c5b7c-ee27-4f94-9ea8-192393746ddd",
   "status": "close",
   "serverUrl": "https://example.evolution-api.com",
   "apikey": "123456",
```

```
    "integration": {
      "token": "123456",
      "webhook_wa_business": "https://example.evolution-api.com/webhook/whatsapp/teste-docs"
    }
  }
 }
]
```


Instance Connect

GET
/
instance
/
connect
/
{instance}
Try it
Authorizations

apikey
string
header
required
Your authorization key header
Path Parameters

instance
string
required
Name of the instance to connect
Query Parameters

number
string
Phone number (with country code) to be connected
Response
200
application/json
Ok

pairingCode
string

The unique code used for pairing a device or account.

code
string
A specific code associated with the pairing process. This may include tokens or other identifiers.

count
integer
The count or number of attempts or instances related to the pairing process.

Instances Connect curl

```
curl --request GET \
  --url https://{server-url}/instance/connect/{instance} \
  --header 'apikey: <api-key>'
```

Instances Connect javascript

```
const url = 'https://{server-url}/instance/connect/{instance}';
const options = {method: 'GET', headers: {apikey: '<api-key>'}, body: undefined};

try {
  const response = await fetch(url, options);
  const data = await response.json();
  console.log(data);
} catch (error) {
  console.error(error);
}
```

200


```
{
  "pairingCode": "WZYEH1YY",
  "code": "2@y8eK+bjtEjUWy9/FOM...",
  "count": 1
}
```


404


```
{
  "status": 404,
  "error": "Not Found",
  "response": {
    "message": [
      "The \"invalid-instance\" instance does not exist"
    ]
  }
}
```

Restart Instance

PUT
/
instance
/
restart
/
{instance}
Try it
Authorizations

apikey
string
header
required
Your authorization key header
Path Parameters

instance
string
required
Name of the instance to restart
Response
200
application/json
Ok

instance
object
Show child attributes

Restart Instance curl

```
curl --request PUT \
  --url https://{server-url}/instance/restart/{instance} \
  --header 'apikey: <api-key>'
```

Restart Instance javascript

```
const url = 'https://{server-url}/instance/restart/{instance}';
const options = {method: 'PUT', headers: {apikey: '<api-key>'}, body: undefined};

try {
  const response = await fetch(url, options);
```

```
  const data = await response.json();
  console.log(data);
} catch (error) {
  console.error(error);
}
```

200

```
{
  "instance": {
    "instanceName": "teste-docs",
    "state": "open"
  }
}
```

404

```
{
  "status": 404,
  "error": "Not Found",
  "response": {
    "message": [
      "The \"invalid-instance\" instance does not exist"
    ]
  }
}
```

Connection State

GET

/

instance

/

connectionState

/

{instance}

Try it

Authorizations

apikey

string

header

required

Your authorization key header

Path Parameters

instance

string

required

Name of the instance to get status connect

Response

200

application/json

Ok

instance

object

Hide child attributes

instance.instanceName

string

The name of the instance.

instance.state

string

The state of the instance.

Connection State curl

```
curl --request GET \
  --url https://{server-url}/instance/connectionState/{instance} \
  --header 'apikey: <api-key>'
```

Connection State javascript

```
const url = 'https://{server-url}/instance/connectionState/{instance}';
const options = {method: 'GET', headers: {apikey: '<api-key>'}, body: undefined};

try {
  const response = await fetch(url, options);
  const data = await response.json();
  console.log(data);
} catch (error) {
  console.error(error);
}
```

200

```
{
  "instance": {
    "instanceName": "teste-docs",
    "state": "open"
  }
}
```

404

```
{
  "status": 404,
  "error": "Not Found",
  "response": {
    "message": [
      "The \"invalid-instance\" instance does not exist"
    ]
  }
}
```


Logout Instance

DELETE

/

instance

/

logout

/

{instance}

Try it

Authorizations

apikey

string

header

required

Your authorization key header

Path Parameters

instance

string

required

Name of the instance to logout

Response

200

application/json

Ok

status

string

The status of the response.

error

boolean

Indicates whether an error occurred.

response
object
Show child attributes
Logout Instance curl

```
curl --request DELETE \
  --url https://{server-url}/instance/logout/{instance} \
  --header 'apikey: <api-key>'
```

Logout Instance javascript

```
const url = 'https://{server-url}/instance/logout/{instance}';
const options = {method: 'DELETE', headers: {apikey: '<api-key>'}, body: undefined};

try {
  const response = await fetch(url, options);
  const data = await response.json();
  console.log(data);
} catch (error) {
  console.error(error);
}
```

200

```
{
  "status": "SUCCESS",
  "error": false,
  "response": {
    "message": "Instance logged out"
  }
}
```

404

```
{
  "status": 404,
  "error": "Not Found",
  "response": {
    "message": [
      "The \"invalid-instance\" instance does not exist"
    ]
  }
}
```

Delete Instance

DELETE
/

instance
/
delete
/
{instance}
Try it
Authorizations

apikey
string
header
required
Your authorization key header
Path Parameters

instance
string
required
Name of the instance to delete
Response
200
application/json
Ok

status
string
The status of the response.

error
boolean
Indicates whether an error occurred.

response
object
Show child attributes

Delete Instance curl

```
curl --request DELETE \
  --url https://{server-url}/instance/delete/{instance} \
  --header 'apikey: <api-key>'
```

Delete Instance javascript

```
const url = 'https://{server-url}/instance/delete/{instance}';
const options = {method: 'DELETE', headers: {apikey: '<api-key>'}, body: undefined};

try {
  const response = await fetch(url, options);
  const data = await response.json();
  console.log(data);
} catch (error) {
  console.error(error);
}
```

200


```
{
  "status": "SUCCESS",
  "error": false,
  "response": {
    "message": "Instance deleted"
  }
}
```


404


```
{
  "status": 404,
  "error": "Not Found",
  "response": {
    "message": [
      "The \"invalid-instance\" instance does not exist"
    ]
  }
}
```


Set Presence

POST

/
instance
/
setPresence
/
{instance}
Try it

Authorizations

apikey
string
header
required

Your authorization key header
Path Parameters

instance
string
required
Name of the instance to connect
Body
application/json

presence
enum<string>
required
Available options:
available,
unavailable

Set Presence curl

```
curl --request POST \
  --url https://{server-url}/instance/setPresence/{instance} \
  --header 'Content-Type: application/json' \
  --header 'apikey: <api-key>' \
  --data '{
  "presence": "available"
}
```

Set Presence javascript

```
const url = 'https://{server-url}/instance/setPresence/{instance}';
const options = {
  method: 'POST',
  headers: {apikey: '<api-key>', 'Content-Type': 'application/json'},
  body: '{"presence":"available"}'
};

try {
  const response = await fetch(url, options);
  const data = await response.json();
  console.log(data);
} catch (error) {
  console.error(error);
}'
```

200

This response has no body data.

404


```
{
  "status": 404,
  "error": "Not Found",
  "response": {
    "message": [
      "The \"invalid-instance\" instance does not exist"
    ]
  }
}
```


Webhook

Set Webhook

POST

/
webhook
/
set
/
{instance}
Try it

Authorizations

apikey
string
header
required
Your authorization key header

Path Parameters

instance
string
required
Name of the instance

Body
application/json

enabled
boolean
required

enable webhook to instance

url
string
required
Webhook URL

webhookByEvents
boolean
required
Enables Webhook by events

webhookBase64
boolean
required
Sends files in base64 when available

events
enum<string>[]
required
Events to be sent to the Webhook
Minimum length: 1
Show child attributes

Response
201 - application/json
Created

webhook
object
Show child attributes

webhook.instanceName
string
The name of the instance.

webhook.webhook
object
Show child attributes

Set Webhook curl

```
curl --request POST \
  --url https://{server-url}/webhook/set/{instance} \
  --header 'Content-Type: application/json' \
  --header 'apikey: <api-key>' \
  --data '{
  "enabled": true,
  "url": "<string>",
  "webhookByEvents": true,
  "webhookBase64": true,
  "events": [
    "APPLICATION_STARTUP"
  ]
}'
```

Set Webhook javascript

```
const url = 'https://{server-url}/webhook/set/{instance}';
const options = {
  method: 'POST',
  headers: {apikey: '<api-key>', 'Content-Type': 'application/json'},
  body:
'{"enabled":true,"url":"<string>","webhookByEvents":true,"webhookBase64":true,"events":["APPL
ICATION_STARTUP"]}'
};

try {
  const response = await fetch(url, options);
  const data = await response.json();
  console.log(data);
} catch (error) {
  console.error(error);
}
```

201

```
{
  "webhook": {
    "instanceName": "teste-docs",
    "webhook": {
      "url": "https://example.com",
      "events": [
        "APPLICATION_STARTUP"
      ],
      "enabled": true
    }
  }
}
```

Find Webhook

GET
/
webhook
/
find
/
{instance}
Try it
Authorizations

apikey
string
header
required
Your authorization key header
Path Parameters

instance
string
required
Name of the instance
Response
200 - application/json
Ok

enabled
boolean
Indicates whether the webhook is enabled.

url
string
The URL of the webhook.

events
string[]
List of events the webhook is subscribed to.

Find Webhook curl

```
curl --request GET \
  --url https://{server-url}/webhook/find/{instance} \
  --header 'apikey: <api-key>'
```

Find Webhook javascript

```
const url = 'https://{server-url}/webhook/find/{instance}';
const options = {method: 'GET', headers: {apikey: '<api-key>'}, body: undefined};

try {
  const response = await fetch(url, options);
  const data = await response.json();
  console.log(data);
} catch (error) {
  console.error(error);
}
```

200

```
{
  "enabled": true,
  "url": "https://example.com",
  "events": [
    "APPLICATION_STARTUP"
  ]
}
```

Settings

Set Settings

POST

/
settings
/
set
/
{instance}
Try it

Authorizations

apikey
string
header
required
Your authorization key header

Path Parameters

instance
string
required
Name of the instance
Body
application/json

rejectCall
boolean
required
Reject calls automatically

msgCall
string
required
Message to be sent when a call is rejected automatically

groupsIgnore
boolean
required
Ignore group messages

alwaysOnline
boolean
required
Always show WhatsApp online

readMessages
boolean
required
Send read receipts

readStatus
boolean
required
See message status

syncFullHistory

boolean
required
Syncronize full WhatsApp history with EvolutionAPI
Response
201 - application/json
Created

settings
object
Show child attributes

settings.instanceName
string
The name of the instance.

settings.settings
object
Show child attributes

Set Settings curl
```
curl --request POST \
  --url https://{server-url}/settings/set/{instance} \
  --header 'Content-Type: application/json' \
  --header 'apikey: <api-key>' \
  --data '{
  "rejectCall": true,
  "msgCall": "<string>",
  "groupsIgnore": true,
  "alwaysOnline": true,
  "readMessages": true,
  "readStatus": true,
  "syncFullHistory": true
}'
```

Set Settings javascript
```
const url = 'https://{server-url}/settings/set/{instance}';
const options = {
  method: 'POST',
  headers: {apikey: '<api-key>', 'Content-Type': 'application/json'},
  body:
'{"rejectCall":true,"msgCall":"<string>","groupsIgnore":true,"alwaysOnline":true,"readMessages":t
rue,"readStatus":true,"syncFullHistory":true}'
};
```

```
try {
  const response = await fetch(url, options);
  const data = await response.json();
  console.log(data);
} catch (error) {
  console.error(error);
}
```

201

```
{
  "settings": {
    "instanceName": "teste-docs",
    "settings": {
      "reject_call": true,
      "groups_ignore": true,
      "always_online": true,
      "read_messages": true,
      "read_status": true,
      "sync_full_history": false
    }
  }
}
```

Find Settings

GET

/

settings

/

find

/

{instance}

Try it

Authorizations

apikey

string

header

required

Your authorization key header

Path Parameters

instance

string

required

Name of the instance to get settings

Response

200 - application/json

Ok

reject_call

boolean

Indicates whether to reject incoming calls.

groups_ignore

boolean

Indicates whether to ignore group messages.

always_online

boolean

Indicates whether to always keep the instance online.

read_messages

boolean

Indicates whether to mark messages as read.

read_status

boolean

Indicates whether to read status updates.

sync_full_history

boolean

Indicates whether to synchronize full message history.

Find Webhook curl

```
curl --request GET \
  --url https://{server-url}/settings/find/{instance} \
  --header 'apikey: <api-key>'
```

Find Webhook javascript

```
const url = 'https://{server-url}/settings/find/{instance}';
const options = {method: 'GET', headers: {apikey: '<api-key>'}, body: undefined};

try {
  const response = await fetch(url, options);
```

```
  const data = await response.json();
  console.log(data);
} catch (error) {
  console.error(error);
}
```

200

```
{
  "reject_call": true,
  "groups_ignore": true,
  "always_online": true,
  "read_messages": true,
  "read_status": true,
  "sync_full_history": false
}
```

Settings

Send Plain Text

POST

/

message

/

sendText

/

{instance}

Try it

Authorizations

apikey

string

header

required

Your authorization key header

Path Parameters

instance

string

required

Name of the instance

Body

application/json

number
string
required
Number to receive the message (with country code)

text
string
required
Test message to send

delay
integer
Presence time in milliseconds before sending message

linkPreview
boolean
Shows a preview of the target website if there's a link within the message

mentionsEveryOne
boolean
Mentioned everyone when the message send

mentioned
enum<string>[]
Numbers to mention
Show child attributes

quoted
object
Show child attributes
Response
201 - application/json
Created

key
object
Show child attributes

message
object
Show child attributes

message.extendedTextMessage
object
Show child attributes

messageTimestamp
string
The timestamp of the message.

status
string
The status of the message.

Send Text curl

```
curl --request POST \
 --url https://{server-url}/message/sendText/{instance} \
 --header 'Content-Type: application/json' \
 --header 'apikey: <api-key>' \
 --data '{
 "number": "<string>",
 "text": "<string>",
 "delay": 123,
 "linkPreview": true,
 "mentionsEveryOne": true,
 "mentioned": [
  "{{remoteJID}}"
 ],
 "quoted": {
  "key": {
   "id": "<string>"
  },
  "message": {
   "conversation": "<string>"
  }
 }
```

```
}'
```

Send Text javascript

```
const url = 'https://{server-url}/message/sendText/{instance}';
const options = {
  method: 'POST',
  headers: {apikey: '<api-key>', 'Content-Type': 'application/json'},
  body:
'{"number":"<string>","text":"<string>","delay":123,"linkPreview":true,"mentionsEveryOne":true,"
mentioned":["{{remoteJID}}"],"quoted":{"key":{"id":"<string>"},"message":{"conversation":"<string
>"}}}'
};

try {
  const response = await fetch(url, options);
  const data = await response.json();
  console.log(data);
} catch (error) {
  console.error(error);
}
```

201


```
{
  "key": {
    "remoteJid": "553198296801@s.whatsapp.net",
    "fromMe": true,
    "id": "BAE594145F4C59B4"
  },
  "message": {
    "extendedTextMessage": {
      "text": "Olá!"
    }
  },
  "messageTimestamp": "1717689097",
  "status": "PENDING"
}
```


Chat controller

Check is WhatsApp

POST

/

chat

/
whatsappNumbers
/
{instance}
Try it
Authorizations

apikey
string
header
required
Your authorization key header
Path Parameters

instance
string
required
Name of the instance
Body
application/json

numbers
string[]
Phone numbers (with country code) to be checked
Response
200 - application/json
Ok
Array of objects representing WhatsApp account existence information.

exists
boolean
Indicates whether the WhatsApp account exists.

jid
string
The JID of the WhatsApp account.

number
string
The phone number associated with the WhatsApp account.

WhatsApp Numbers curl

```
curl --request POST \
  --url https://{server-url}/chat/whatsappNumbers/{instance} \
  --header 'Content-Type: application/json' \
  --header 'apikey: <api-key>' \
  --data '{
  "numbers": [
    "<string>"
  ]
}'
```

WhatsApp Numbers javascript

```
const url = 'https://{server-url}/chat/whatsappNumbers/{instance}';
const options = {
  method: 'POST',
  headers: {apikey: '<api-key>', 'Content-Type': 'application/json'},
  body: '{"numbers":["<string>"]}'
};

try {
  const response = await fetch(url, options);
  const data = await response.json();
  console.log(data);
} catch (error) {
  console.error(error);
}
```

200

```
[
  {
    "exists": true,
    "jid": "553198296801@s.whatsapp.net",
    "number": "553198296801"
  }
]
```

Mark Message As Read

POST

/

chat

/

markMessageAsRead

/

{instance}

Try it

Authorizations

apikey
string
header
required
Your authorization key header
Path Parameters

instance
string
required
Name of the instance
Body
application/json

readMessages
object[]
required
Messages to be mark as read
Show child attributes

Response
201 - application/json
Created

message
string
A brief message describing the action performed.

read
string
The status of the read action.

Mark Message As Read curl

```
curl --request POST \
  --url https://{server-url}/chat/markMessageAsRead/{instance} \
  --header 'Content-Type: application/json' \
  --header 'apikey: <api-key>' \
  --data '{
  "readMessages": [
    {
      "remoteJid": "<string>",
```

```
    "fromMe": true,
    "id": "<string>"
   }
 ]
}'
```

Mark Message As Read javascript

```
const url = 'https://{server-url}/chat/markMessageAsRead/{instance}';
const options = {
 method: 'POST',
 headers: {apikey: '<api-key>', 'Content-Type': 'application/json'},
 body: '{"readMessages":[{"remoteJid":"<string>","fromMe":true,"id":"<string>"}]}'
};

try {
 const response = await fetch(url, options);
 const data = await response.json();
 console.log(data);
} catch (error) {
 console.error(error);
}
```

201

```
{
 "message": "Read messages",
 "read": "success"
}
```

Mark Message As Unread

POST

/
chat
/
markChatUnread
/
{instance}
Try it

Authorizations

apikey
string
header
required

Your authorization key header
Path Parameters

instance
string
required
Name of the instance
Body
application/json

lastMessage
object[]
required
Messages to be mark as unread
Show child attributes

chat
string
required
remoteJid here
Response
201 - application/json
Created

message
string
A brief message describing the action performed.

read
string
The status of the read action.

Mark Message As Unread curl

```
curl --request POST \
  --url https://{server-url}/chat/markChatUnread/{instance} \
  --header 'Content-Type: application/json' \
  --header 'apikey: <api-key>' \
  --data '{
  "lastMessage": [
    {
      "remoteJid": "<string>",
```

```
    "fromMe": true,
    "id": "<string>"
   }
 ],
 "chat": "<string>"
}'
```

Mark Message As Unread javascript

```
const url = 'https://{server-url}/chat/markChatUnread/{instance}';
const options = {
 method: 'POST',
 headers: {apikey: '<api-key>', 'Content-Type': 'application/json'},
 body:
'{"lastMessage":[{"remoteJid":"<string>","fromMe":true,"id":"<string>"}],"chat":"<string>"}'
};

try {
 const response = await fetch(url, options);
 const data = await response.json();
 console.log(data);
} catch (error) {
 console.error(error);
}
```

201

```
{
 "message": "Read messages",
 "read": "success"
}
```

Archive Chat

POST

/
chat
/
archiveChat
/
{instance}
Try it

Authorizations

apikey
string
header

required
Your authorization key header
Path Parameters

instance
string
required
Name of the instance
Body
application/json

lastMessage
object
required
Messages to be mark as read
Show child attributes

archive
boolean
required
Whether to archive the chat or not

chat
string
required
remoteJid here
Response
201 - application/json
Created

chatId
string
The ID of the chat.

archived
boolean
Indicates whether the chat is archived.

Archive Chat curl
curl --request POST \

```
 --url https://{server-url}/chat/archiveChat/{instance} \
 --header 'Content-Type: application/json' \
 --header 'apikey: <api-key>' \
 --data '{
 "lastMessage": {
  "key": {
   "remoteJid": "<string>",
   "fromMe": true,
   "id": "<string>"
  }
 },
 "archive": true,
 "chat": "<string>"
}'
```

Archive Chat curl

```
const url = 'https://{server-url}/chat/archiveChat/{instance}';
const options = {
 method: 'POST',
 headers: {apikey: '<api-key>', 'Content-Type': 'application/json'},
 body:
'{"lastMessage":{"key":{"remoteJid":"<string>","fromMe":true,"id":"<string>"}},"archive":true,"cha
t":"<string>"}'
};

try {
 const response = await fetch(url, options);
 const data = await response.json();
 console.log(data);
} catch (error) {
 console.error(error);
}
```

201

```
{
 "chatId": "553198296801@s.whatsapp.net",
 "archived": true
}
```

Chat controller

Fetch Business Profile

POST
/
chat
/
fetchBusinessProfile
/
{instance}
Try it
Authorizations

apikey
string
header
required
Your authorization key header
Path Parameters

instance
string
required
Name of the instance
Body
application/json

number
string
required
Phone number with country code

Fetch Business Profile curl

```
curl --request POST \
  --url https://{server-url}/chat/fetchBusinessProfile/{instance} \
  --header 'Content-Type: application/json' \
  --header 'apikey: <api-key>' \
  --data '{
  "number": "<string>"
}'
```

Fetch Business Profile javascript

```
const url = 'https://{server-url}/chat/fetchBusinessProfile/{instance}';
const options = {
  method: 'POST',
  headers: {apikey: '<api-key>', 'Content-Type': 'application/json'},
  body: '{"number":"<string>"}'
```

```
};

try {
  const response = await fetch(url, options);
  const data = await response.json();
  console.log(data);
} catch (error) {
  console.error(error);
}
```

Fetch Profile

POST
/
chat
/
fetchProfile
/
{instance}
Try it
Authorizations

apikey
string
header
required
Your authorization key header
Path Parameters

instance
string
required
Name of the instance
Body
application/json

number
string
required
Phone number with country code