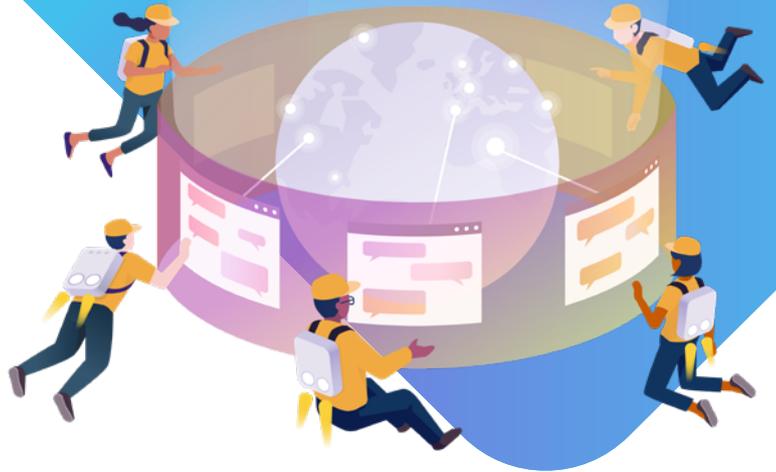




Mobile Chat

Protocol: REST API



Interface information

Send a message with "Hello World!"

```
curl -u username:password -X POST -H "Content-Type: application/json" -d '{"from":"MyCompany",  
"to":"+491622943176", "message":"Hello World!"}' https://rest.tyntec.com/messaging/im/v1/outbound/requests
```

The REST API (Representational State Transfer Application Programming Interface) enables quick and easy access to tyntec's messaging platform. The communication is established via HTTPS (Hypertext Transfer Protocol Secure) connections. The exchanged object types are JSON (JavaScript Object Notation). tyntec accepts "application/json" as "Content-Type" (and "charset=utf-8") in the HTTP header.

The Mobile Chat API service is accessible through the base URL (Unique Resource Location) (`{baseURL}`):

```
https://rest.tyntec.com/messaging/im/v1/
```

tyntec will provide a username (`{username}`) and password (`{password}`) to grant access to the services:

```
https://{username}:{password}@rest.tyntec.com/messaging/im/v1/
```

The Base64-encoded combination "`{username}:{password}`" is used as an authentication-token for the Basic HTTP authentication and can also be given in the HTTP header:

```
Authorization: Basic {authentication-token}
```

Send SMS messages

Resource URLs for sending messages

The Mobile Chat API service is governed by two operations that allow you to send a message and to query the status of a previously sent message (tyntec stores the status details for up to three months after the final delivery state is known).

Operation	Resource URL	Method
Send a message	<code>\${baseUrl}/outbound/requests</code>	POST
Query the status of a message	<code>\${baseUrl}/outbound/requests/{requestId}/status</code>	GET
Receive a message	<code>\${baseUrl}/inbound/requests</code>	POST

Connection Steps

Step 1: Request "Send a message"

To send a new, outgoing message, make a POST request to the request's resource. The following code blocks give a basic example of how to send a message:

Basic Example for Sending a Message (JSON)

```
curl
-u username:password
-X POST
-H "Content-Type: application/json"
-d '{
  "from": "MyCompany",
  "to": "+491622943176",
  "message": "This is an example."
}'
https://rest.tyntec.com/messaging/im/v1/outbound/requests
```

Response to Basic Example for Sending a Message (JSON)

```
HTTP/1.1 200 OK
{
  "requestId": "12-75cfafba-8799-4f2b-8505-ece6af6abf01",
  "errorText": ""
}
```

In this request, there are a number of parameters that need to be defined, while others are optional:

Parameter	Optional/ Mandatory	Possible Values	Description
from	mandatory	any international phone number OR short code OR alphanumeric sender ID (1-11 characters)	This parameter gives the identification of the sending party, which can either be the phone number in international format or a short code, or an alphanumeric identifier with up to 11 characters. When the message is sent to a messaging app, a predefined name identifying your service may be used instead of this field. When the message is sent via SMS, some destination networks impose restrictions on the sender ID format. Please check the coverage list and/or contact your account manager for more information.

Parameter	Optional/ Mandatory	Possible Values	Description
to	mandatory	any international phone number	This parameter represents the receiving party's phone number in international format.
message	optional	any UTF-8 encoded string	The message will be delivered as such to messaging apps. When delivered via SMS, the maximum length is 70 characters per single message and 63 characters per concatenated message.
imageUrl	optional	any valid URL maximum 62 characters long	If this parameter is present, the image URL is sent to the messaging app and if supported by it, displayed. If the message falls back to SMS, the URL is included in the message.
buttonActionUrl	optional	any valid URL maximum 62 characters long	This parameter controls which action should be executed, when the recipient clicks on the button.
buttonCaption	optional	any UTF-8 encoded string	This parameter controls which text is displayed on the button in a messaging app.
imChannels	optional	"FACEBOOK", "VIBER", "FACEBOOK, VIBER", "VIBER, FACEBOOK"	This parameter enables to specify the IM channels the system will attempt to deliver the messages to and in which order. It can be "FACEBOOK", "VIBER", "FACEBOOK , VIBER" or "VIBER , FACEBOOK". The system will attempt to deliver to the next messaging app only in case the message cannot be delivered to the previous one. If not specified, the default order is "VIBER , FACEBOOK". Please note that the SMS fallback is controlled by "deliverOnlyVialM" parameter.
deliverOnlyVialM	optional	true/false	If this parameter is set to "true", the fallback to SMS is deactivated. The message will be attempted only via IM and will fail if it cannot be delivered to one of the supported messaging apps.
ttl	optional	any positive integer value	The message validity period in seconds. While tyntec does not impose a minimum ttl, some messaging apps might ignore a ttl value lower than a certain threshold. ttl range: 30 - 86400 seconds. If it is not set, the default is 14 days. Supported by: Viber, SMS
ttlExpiryText	optional	any UTF-8 encoded string	If this parameter is defined and supported by the destination messaging app, the user will see this text instead of the original text defined in the message parameter in case the message is delivered after the expiration of the ttl. Supported by: Viber *Relevant for iOS only. For Android, if the message was not delivered before the time ended, the message will not appear.
callbackUrl	optional	any UTF-8 encoded string	We will use this URL to submit the delivery receipts to you. The following scenarios are possible: 1. You have the option to define the callback URL as a global parameter by asking tyntec Support. If you do this, you don't need to specify the callback URL in each request. If you do specify it in the request though, it overrides the default. 2. You can choose not to define a global callback URL, and then you need to specify it in every request. 3. If you don't define the global URL and you also don't define the callbackURL parameter, tyntec will not send the delivery receipts to you.
messagePurpose	mandatory	"promotion" or "transaction"	This parameter is mandatory when sending messages to Viber.

		Viber
Message length (characters)		1000
Button caption length (characters)		30
Message types	Text only	Yes
	Image only	Yes
	Text + image	No
	Text + button	Yes
	Image + button	No
	Text + image + button	Yes

Step 2: Response to request "Send a message"

tyntec will immediately respond to a HTTPS POST request to "Send a message". The response will contain the request ID. In the case there were delivery issues, the response will contain possible delivery issues following an immediate check for inconsistencies (e.g. case sensitivity; wrong parameter; too many characters; no receiver defined; etc.). The corresponding HTTP status code will be returned.

Successful Request Response (JSON)

```
HTTP/1.1 200 OK
{
  "requestId":"12-75cfafba-8799-4f2b-8505-ece6af6abf01",
  "errorText":""
}
```

Unsuccessful Request Response (JSON)

```
HTTP/1.1 400 Bad Request
{
  "requestId":"12-75cfafba-8799-4f2b-8505-ece6af6abf01","responseText":"*see list below*"
}
```

HTTP Code	Message	Description
200 OK	*see code examples 'Un/Successful Request Response'* OR *see code examples 'Message Notification'*	The HTTPS request was accepted. The posted message passed an initial consistency check and tyntec will reply with the respective requestId. OR The requested message notification can be retrieved and is returned.
400 Bad Request	*see validation error message list below*	The HTTPS request failed to pass the validation, i.e., one or more of the required parameters of the request are empty/missing/contradictory/invalid. The given message provides more information as to why the validation failed.
403 Forbidden	Username/password wrong or missing.	The credentials provided are not authorized.
404 Not Found	The requested message notification cannot be retrieved.	The requested URI is either nonexistent or the respective authentication is invalid.
405 Method Not Allowed	Invalid request method.	The request contains an invalid HTTP method (POST/GET).
406 Not Acceptable	The content-type is not supported. Allowed values are "application/json".	The request is not JSON format or the HTTP header reads "Content-Type".
415 Unsupported Media Type	The media type is not supported. Allowed media-type is "application/json".	The request is not JSON format or the HTTP header reads "Accept".
500 Internal Server Error	Please retry. If error persists, please contact customer support.	Unexpected error.

Validation error message may be a one or any combination of the ones below:

Message	Sent	Description
Invalid authentication.	Send/Query actions	Invalid username and/or password, or wrong authentication token.
RequestId is missing.	Query action	Please specify the request-ID for which the message status should be retrieved.
Sender is missing.	Send action	The mandatory parameter "from" is not specified.
Receiver is missing.	Send action	The mandatory parameter "to" is not specified.
Receiver number is not numeric.	Send action	The mandatory parameter "to" is malformed. It has to be an international phone number.
Message is empty.	Send action	An empty message cannot be sent.
Message exceeds maximum character length of 1000.	Send action	Message text exceeds the limit of 1000 characters.
The respond-back URL is invalid.	Send action	Please format the URL according to: http(s)://host(:port).
One or more mandatory parameters are missing.	Send action	The request is missing mandatory parameters.
Your message is rejected because it has over-length.	Send action	Your account settings block the sending of over-length messages.
The requested URI was not found.	Query action	The requested URI is either nonexistent or the respective authentication is invalid.
Unexpected error, please retry. If error persists, please contact customer support.	Send/Query actions	Something unexpected occurred during the processing of the request.
The parameter \${invalidparameter} is invalid.	Send action	The request contains at least one invalid parameter. \${invalidparameter} will state which parameter.

Step 3: Query the status of a sent message / message notification

The tyntec server will respond to your query with the current message notification. tyntec stores message notifications for three months. tyntec will also POST these message notifications in case there is a call-back URL specified.

Query the Status of a Request/Message Notification (JSON)

```
curl https://username:password@rest.tyntec.com/messaging/im/v1/outbound/requests/12-75cfafba-8799-4f2b-8505-ece6af6abf01/status
```

Message Notification – Delivery via Instant Messaging (JSON)

```
{
  "deliveryChannel": {
    "channel": "IM",
    "ott": "Viber"
  },
  "from": "MyCompany",
  "overallState": "DELIVERED",
  "requestId": "12-75cfafba-8799-4f2b-8505-ece6af6abf01",
  "sendParts": [
    {
      "contentExcerpt": "http://www.example.c",
      "currency": "EUR",
      "deliveryState": "DELIVERED",
      "deliveryStateDate": 1454066954667,
      "errorCode": "",
      "messagePartId": 1
    }
  ],
  "size": 1,
  "to": "+4917912345678",
  "userContext": null
}
```

Message Notification – Delivery via SMS (JSON)

```
{
  "deliveryChannel": {
    "channel": "SMS",
    "mccmnc": null,
    "ttid": "16"
  },
  "from": "MyCompany",
  "overallState": "DELIVERED",
  "requestId": "12-75cfafba-8799-4f2b-8505-ece6af6abf01",
  "sendParts": [
    {
      "contentExcerpt": "http://www.example.c",
      "currency": "EUR",
      "deliveryState": "DELIVERED",
      "deliveryStateDate": 1454066954667,
      "errorCode": "",
      "messagePartId": 1,
      "price": 0.5,
      "priceEffective": 1454067015769,
      "sentDate": 1454066949962
    },
    {
      "contentExcerpt": "This is a test.",
      "currency": "EUR",
      "deliveryState": "DELIVERED",
      "deliveryStateDate": 1454066954631,
      "errorCode": "",
      "messagePartId": 2,
      "price": 0.5,
      "priceEffective": 1454067015769,
      "sentDate": 1454066949994
    }
  ],
  "size": 2,
  "to": "+4917912345678",
  "userContext": null
}
```

Parameter	Description
requestId	The unique identifier provided for each request.
overallState	Aggregated status from all respective deliveryStatus.
size	1 if the message is delivered via IM or the amount of concatenated parts if the message is delivered via SMS.
from	This parameter gives the identification of the sending party, which can either be the phone number in international format or an alphanumeric identifier with up to 11 characters.
to	This parameter represents the receiving party's phone number in international format.
userContext	The custom message context provided by the customer.
deliveryChannel	Specifies if the message was delivered to a messaging app ("IM") or via SMS ("SMS").
ott	The name of the messaging app to which the message was delivered.
mccmnc	A representative IMSI prefix of the target network. The respective mapping can be found at http://www.mcc-mnc.com/ , for example.
ttid	The respective tyntec ID of the target network.
overallPrice	The aggregated price from all respective message parts.
MessageNotificationContent	The actual message notifications for respective parts.
message	The separator for each part.
deliveryState	The delivery status of this specific part. Possible values are given in a following table.
contentExcerpt	The first 20 characters of the sent message.
messagePartId	A number identifying the message part.
price	The price for sending the respective message part.
currency	The currency in which the price is given, corresponding to the currency of the invoice.
priceEffective	The date when the "price" became active.
sentDate	The time stamp when the message was sent out by tyntec for delivery.
doneDate	The time stamp when the message was successfully delivered.
errorCode	The reason for an unsuccessful delivery attempt. Possible values are provided in the following table .

Possible values for "deliveryStatus":

Parameter	Description
DELIVERED	Message was successfully delivered.
SEEN	Message was seen by the recipient on their device. (Not supported by Viber anymore)
UNDELIVERABLE	Message cannot be delivered. See error codes for more details.
EXPIRED	Message expired, i.e., the message could not be delivered within the specified time span. See error codes for more details.
REJECTED	Message rejected, i.e., the message could not be accepted by our platform. See error codes for more details.
BUFFERED	Message could not be delivered. tyntec is in the process of resending. Please wait for one of the above final statuses.

Receive messages

Receive a message

This request forwards incoming messages from users to you via an HTTPS request to your webserver. For this service, you will need to provide tyntec with the URL (\$callbackURL) of your webserver, e.g.

<https://rest.customer.com/inboundmessages>

If such a URL is not provided, tyntec will forward the inbound messages to the callback URL you specify to be used for returning the status notifications to you.

Receive an inbound message

To forward a new incoming message, tyntec makes a POST request to your \$callbackURL. tyntec's application will retry delivery every 10 seconds in case your application does not accept the inbound message (200, 201, 202 status code). Retries are paused for 10 minutes after every 100 consecutive unsuccessful delivery attempts. tyntec's application will retry for a maximum of 48 hours.

The following code block gives a basic example of how the request will look like:

Receiving messages (JSON):

```
curl
-X POST
-H "Content-Type: application/json"
-d
'{
  "from":"+491622943176",
  "to":"customer_account",
  "message":"This is an example.",
  "sentDate":"1454066954667",
  "requestId":"12-75cfafba-8799-4f2b-8505-ece6af6abf01",
  "originalRequestId":"f595e75b-2808-4def-883c-82913b7f8365",
}'
https://rest.customer.com/inboundmessages/
```

Parameter	Possible Values	Description
from	international phone number	The phone number of the user sending the message
to	account name	The name of the account that you use for receiving incoming messages
message	UTF-8 encoded string	The message sent by the user
sentDate	long integer	The timestamp showing when the message was submitted by the user
requestId	UTF-8 encoded string	The unique identifier provided for each request (retries will use the same requestId as the original request for easy message tracking)
originalRequestId	UTF-8 encoded string	The unique identifier of the outbound message which generated this user reply
ott	UTF-8 encoded string	The name of the messaging app from which the message comes