ADVANCED MESSAGING PLATFORM

API Documentation

V1.3.5

AMP API v1.3.5 Copyright © 42 Telecom Ltd. All Rights Reserved October 2026.

fortytwo

CONTENTS

1.Introduction 3

1.1.How does it work? 3

2.Overview 5

3.Authentication 5

3.1.HTTP Headers 5

4.Endpoints 6

4.1.Send Message 6

4.1.1.Request 6

4.1.2.Response 15

4.1.3.Example 17

4.2.Check Message Status 26

4.2.1.Request 26

4.2.2.Response 26

4.2.3.Example 27

5.Callbacks 28

5.1.Whitelist callback server 28

5.2.SSL Support 29

5.3.Callback failure 29

5.4.Message status callback 29

5.5.Message reply callback 31

5.6.Server Example 32

6.References 33

6.1.HTTP Status Code 33

1. INTRODUCTION

An API that offers developers the flexibility to send SMS, RCS Messages as well as IM Messages *(Fortytwo's Broadcast Messenger, Telegram, Whatsapp)* with an easy to use REST API.

SMS

Fortytwo has over a decade of experience in delivering A2P SMS worldwide. Advantages of using the SMS Gateway:

- Global – worldwide network coverage.
- Ubiquitous – SMS can be delivered to every mobile device.
- Unparalleled open rate – 90% of SMS are opened within 3 minutes of delivery.
- Flexible – an internet connection is not required for an SMS to terminate on a mobile phone.
- Industry leading – our proprietary systems are maintained by our in-house engineers.

Instant Message

Fortytwo has partnered with Instant Messaging (IM) / Chat providers to deliver instant messages to your customers. Advantages of using the IM Gateway:

- Innovative – a new and inventive way to reach your audience
- Convenient – fast, easy to use and flexible
- Expansive communication – can include a greater number of characters than SMS
- Interactive – allows for easy 2-way communication with your customers
- Feature-rich – supports content such as text, images and call-to-action buttons.

Pricing

For more information about pricing for messages per country, please click here.

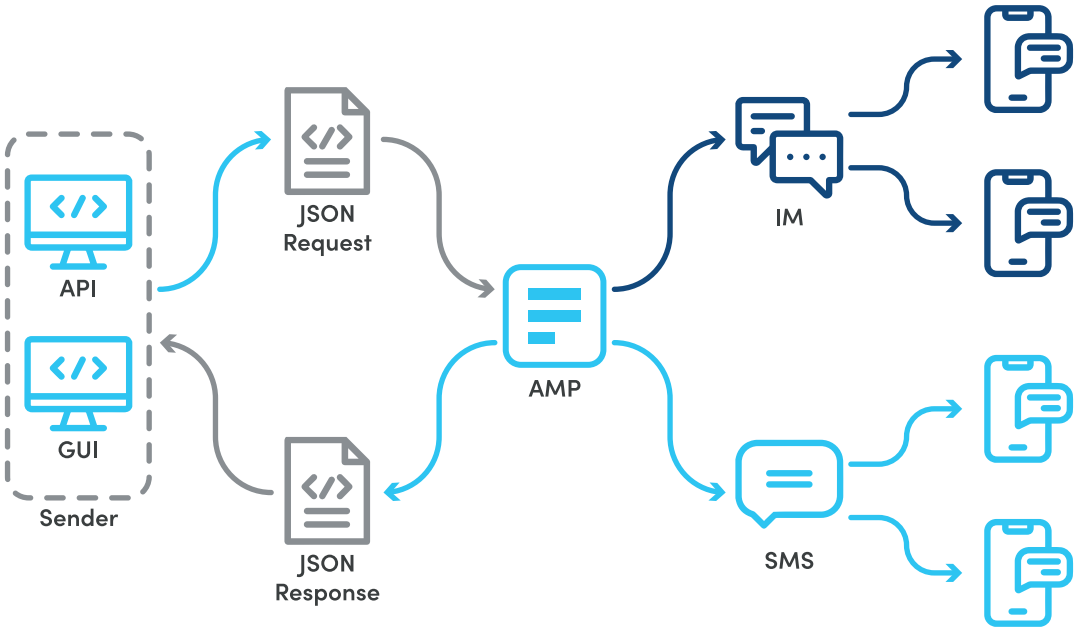
1.1. How does it work?

Fortytwo's AMP is a unified messaging solution platform which conveniently gives you access to SMS and Instant Messaging (IM), all through one API. Our API

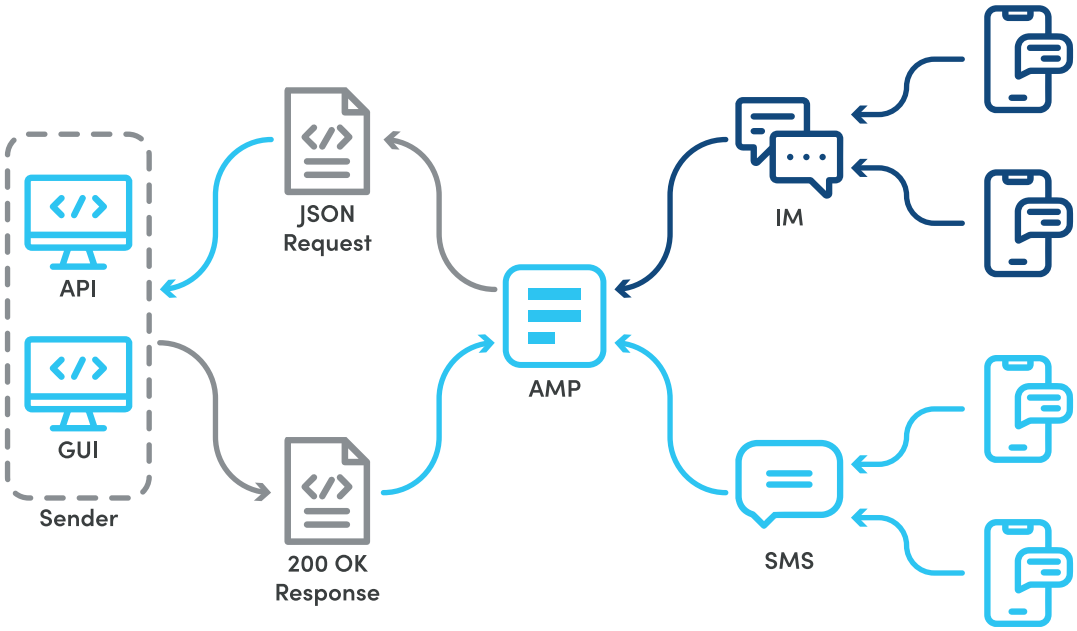
- allows you to set preferences for broadcasting your messages based on your communication preferences and knowledge of your audience
- intelligently determines the most cost-effective delivery of your messages if specific preferences are not defined
- optimises delivery and gives you access to reports on results

With Fortytwo's AMP, you can deliver your messages efficiently, reliably and at competitive rates directly to your customers' mobile phones.

AMP Requests



AMP Callbacks



2. OVERVIEW

To get started, you need to login to our **Control Panel** section to create an authentication token, and use that token within the HTTP headers for every request.

The IM REST web service expects a JSON request that defines the message you wish to send. Your request may include your routing preference, whether it is dependent on cost or on features. The service replies with a JSON response that acknowledges your request, including an identifier for your reference. A call-back service is also available that informs you of the delivery status of a message once it has been processed.

3. AUTHENTICATION

Passing Authentication Token with every request

To make use of the REST API, you must supply an authorisation token in the HTTP header, with each request. The token is generated through the Client Control Panel (**https://controlpanel.fortytwo.com**/), in the tokens section, under the IM tab.

- Each token generated maps directly to your account, message route, sender IDs and IP restrictions. Each token can also have a custom name to make it easier for you to remember what each token is for.
- A set of IPs can be associated to a token, which restricts requests using that token to be sent exclusively from those IP's. If no IP's are specified, your token can be used from anywhere on the Internet.
- Your account can have multiple tokens.
- Tokens do not expire, and can be edited at any time.
- Make sure not to disclose any of your tokens to any unauthorised entity. If this happens, tokens can be invalidated (deleted) from the same user interface. Once a token is invalidated, it cannot be re-used again, instead a new one has to be generated.

3.1. HTTP Headers

Key	Value
Authorization	Token bcdd900c-79f3-4b8d-8bbc-6efXXXXXXXXX
Content-Type	application/json; charset=utf-8

4. ENDPOINTS

4.1. Send Message

The most commonly used feature is the "Send Message" functionality in which the user can send out SMS or IM Messages.

```
POST  https://rest.fortytwo.com/1/xxxx(please speak to account manager for endpoint)
```

4.1.1. Request

Request Body

Key	Type	Required	Description
destinations	DESTINATION_INFO[]	Y	An array of DESTINATION_INFO (see DESTINATION_INFO). Minimum of 1 item in this array. Maximum of 50,000 items in this array.
job_id	String	N	A unique identifier provided by yourself for reference. Maximum 255 characters
message	UBIQUITY_CONTENT	C	Defines a request making use of the ubiquity functionality. When using the ubiquity functionality AMP will attempt to deliver the message at the lowest possible cost via all available channels configured in the API Token being used.When using the ubiquity functionality delivery channel routing is handled internally by AMP and thus all *_content (e.g. sms_content and im_content) sections MUST be omitted. see UBIQUITY_CONTENT
sms_content	SMS_CONTENT	C	Defines the content of your message if sent via SMS.This section MUST be populated if you wish to include SMS as a possible delivery channel for this request.NOTE: if you are making use of the ubiquity functionality (i.e. 'message' section mentioned above) this section MUST be omitted. see SMS_CONTENT
im_content	IM_CONTENT[]	C	Defines the content of your message if sent via IM.This section MUST be populated if you wish to include IM as a possible delivery channel for this request.NOTE: if you are making use of the ubiquity functionality (i.e. 'message' section mentioned above) this section MUST be omitted. see IM_CONTENT
message_plan	Enum	N	Used only when im_content field has 1 or more records (Requiredly). Allowed values are: Enum: FEATURE_RICH — Will attempt to deliver via IM first Enum: LOW_COST — Will attempt the cheapest mode of delivery per destination first Defaults to FEATURE_RICH if not specified.

Key	Type	Required	Description
callback_url	String	N	If set, the callback (delivery / status report) will be delivered to this URL, otherwise no callback will take place. Must be a valid URL starting with http:// or https://. If https is used a valid signed certificate must be used.
reply_url	String	C	If set, any replies sent by the recipient of this message will generate a message-reply callback that will be delivered to the URL given by this field. Note that for this feature to be used, 2-way messaging must be enabled on the Sender ID specified n this message. To enable 2-way messaging on a sender ID, please login to the control panel on the Fortytwo website, or ask for assistance. Must be a valid URL starting with http:// or https://. If https is used a valid signed certificate must be used.
promotional	Boolean	N	If set to true, the message will be flagged as containing promotional content. When not present, it is defaulted to false. A promotional Message is a Message which contains any information that promotes the User's business agenda, the User's business offer or commercial information. The first Message sent to a recipient cannot be promotional, but must be personal, informative and a targeted Message. The user must ensure that all promotional Messages are tagged "Promotional" in the REST API request.

```
{
	"destinations": [{
		. . .
	}],
	"job_id": "",
	"sms_content": {
		. . .
	},
	"im_content": [{
		. . .
	}],
	"message_plan": "",
	"callback_url": "",
	"reply_url": "",
	"promotional": true
}
```

Destination Info (DESTINATION_INFO)

Key	Type	Required	Description
number	String	Y	MSISDN to deliver the message to. Number must be in international format and can only be between 7-15 digits long. First digit cannot be a 0. All numbers must be unique within the "destinations" array.
custom_id	String	N	Your reference for each individual destination.
params	Key-Values	C	If your message content contains placeholders for personalised messages per destination, this field is required to populate the value for each recipient. For further details, see the SMS_CONTENT or IM_CONTENT tables below. For an example on how to use this feature, see Samples.

```
{
        . . .
      "destinations": [{
            "number": "35688000000",
            "custom_id": "50001",
            "params": {
                  "NAME": "Mr John White",
                  "COUNTRY": "Malta"
            },
          . . .
      }]
}
```

Ubiquity Content (UBIQUITY_CONTENT)

Key	Type	Required	Description
texts	TEXT[]	C	Contains a list of text objects. For the time being only one item is allowed in this array. This is required only if all the other fields in this block are not defined. See TEXT.
images	IM_IMAGE[]	C	*See note on allowed media content combinations below this table.Array of objects containing the images of the IM message to send, if message includes images. .For the time being, the maximum size of the array is 1, as only one image is supported per message on the current providers. This is required only if all the other fields in this block are not defined. See IM_IMAGE

Key	Type	Required	Description
actions	IM_ACTION[]	C	*See note on allowed media content combinations below this table.Array of objects describing the action buttons of the IM message to send, if message includes action buttons. For the time being, the maximum size of the array is 1, as only one button is supported per message on the current channels. Action buttons may be rendered differently depending on the destination IM platform. This is required only if all the other fields in this block are not defined. See IM_ACTION.

When using the ubiquity functionality, AMP will attempt to deliver the message to all destinations at the lowest possible cost via all available channels configured in the API Token being used.

For an API Token to support SMS when using the ubiquity functionality, the following parameters need to be set-up when configuring the API Token.

- SMS sender ID
- SMS route

For an API Token to support IM messages when using the ubiquity functionality, the following parameter needs to be set-up when configuring the API Token.

- IM Sender ID

SMS messages only support messages of type TEXT[], a request containing either an IM_IMAGE[] or an IM_ACTION[] block will automatically exclude the SMS channel from the available channels, such messages will only attempt delivery via IM.

```
{
    . . .
    "message": {
        "texts": [{
            . . .
        }],
      "images": [{
            . . .
      }],
      "actions": [{
          . . .
      }]
     }
     . . .
}
```

SMS Content (SMS_CONTENT)

<table>
<tr><th>Key</th><th>Type</th><th>Required</th><th>Description</th></tr>
<tr><td>message</td><td>String</td><td>Y</td><td>String containing the body of the SMS message to be delivered to the destination device. One can Requiredly specify keys within the message body to replace later with values given by the params field in the DESTINATION_INFO object.

In the DESTINATION_INFO section for more information on supplying the value for these keys. Each placeholder must be specified as {#KEY}, where KEY is a key name in the params list. For this feature to be used, GSM7 or UCS2 encoding must be used.

In case of binary (see “encoding” below), this field’s value should be base64 encoded. Minimum length of 1 character. Maximum length is of 5 pages in GSM7 encoding equivalent. When parameter names are used, the maximum length of a message pertains to the resultant message after the parameters have been evaluated. In case of binary message, maximum length is 140 bytes, including the field (see below).</td></tr>
<tr><td>encoding</td><td>Enum</td><td>N</td><td>Specifies how the message will be encoded when forwarding it to the destination device. If unspecified, it will default to GSM7. SMS Character limits:

GSM7 - 160 Char per SMS (153 for concatenated messages)
UCS2 - 70 Char per SMS (67 for concatenated messages)

Message concatenation happens when processing the messages depending on the character count submitted. For more info about concatenation please contact your account manager.

<table>
<tr><th>Enum</th><th>Description</th></tr>
<tr><td>GSM7</td><td>Value of “message” will be converted from UTF-8 (as per HTTP request) to GSM7 prior to forwarding to destination.</td></tr>
<tr><td>UCS2</td><td>Value of “message” will be converted from UTF-8 (as per HTTP request) to UCS2 prior to forwarding to destination.</td></tr>
<tr><td>BINARY</td><td>Value of “message” must contain the binary being sent, encoded using base64.</td></tr>
</table>
</td></tr>
<tr><td>route</td><td>String</td><td>N</td><td>The SMS route to deliver on. If unspecified, it will use the route configured with the authorization token you use (default: G42 -standard global coverage). The only valid values are those routes available for your account.</td></tr>
<tr><td>sender_id</td><td>String</td><td>N</td><td>The sender ID (a.k.a. from) for the SMS delivery. If unspecified, the sender ID tied to your authorization token will be used.If numeric, minimum length is 1, maximum length is 15 digits. If alphanumeric, minimum length is 1, maximum length is 11 characters.</td></tr>
</table>

Key	Type	Required	Description
udh	String	C	User Data Header. Value is in binary, encoded using **base64**. This is required if encoding is set to binary.
pid	int	N	Known as "TP-Protocol-Identifier" (TP-PID). Minimum value: 0 Maximum value: 255 Default: 0 See http://www.3gpp.org/ftp/specs/archive/23_series/23.040/ for further information regarding this field.
ttl	int	N	If supplied, this sets the length of time, in seconds, for which Fortytwo will attempt to send the message on the SMS channel. Note that SMS messages may still be delivered beyond this timeframe, as not all network operators will honour this time value. Value type: 32-bit integer (number of seconds) **Default:** (when not supplied, or when set to 0): Up to route or network defaults. **Min value:** 0 (same as default). **Max value:** 172800 (2 days). Setting a value greater than the maximum value will result in an error.

```
{
    . . .
    "sms_content":{
"message":"VDNKa1pYSWdiblZ0WW1WeUlDTXpNekk0SUdseklISmxZV1I1SUdadmN
pQndhV05yTFhWd0xpQlNaV2RoY21SekxDQkRiMjF3WVc1NUlFRXU=",
      "encoding":"BINARY",
      "validity_period":86400,
      "udh":"VEVTVFVESA==",
      "pid":4,
      "ttl": 0
   },
   . . .
}
```

IM Content (IM_CONTENT)

Key	Type	Required	Description
channel	Enum	N	Specifies the IM provider to use. Possible values: MESSENGER, TELEGRAM, RCS, WHATSAPP
sender_id	String	N	The sender ID (a.k.a. from) for the IM delivery. Sender IDs must be requested via the Fortytwo Control Panel (https://controlpanel.fortytwo.com/usercontrol/Message_services/IM/Sender_IDs/) and be pre-approved by your Fortytwo account manager. If unspecified, the sender ID linked to your authorization token will be used.

Key	Type	Required	Description
content	String	C	*See note on allowed media content combinations below this table. String containing the text of the IM message to send, if message includes text. One can Requiredly specify keys within the message body to replace later with values given by the params field in the DESTINATION_INFO object. See 'params' in the DESTINATION_INFO section for more information on supplying the value for these keys. Each placeholder must be specified as {#KEY}, where KEY is a key name in the params list. **Minimum length:** 1 character **Maximum length:** 1000 characters When parameter names are used, the maximum length of a message pertains to the resultant message after the parameters have been evaluated.
images	IM_IMAGE[]	C	*See note on allowed media content combinations below this table. Array of objects containing the images of the IM message to send, if message includes images. See IM_IMAGE for full description.
actions	IM_ACTION[]	C	*See note on allowed media content combinations below this table. Array of objects describing the action buttons of the IM message to send, if message includes images. See IM_ACTION for full description. Action buttons may be rendered differently depending on the destination IM platform.
ttl	Int	N	If supplied, this sets the length of time, in seconds, for which Fortytwo will attempt to send the message on this channel.**Value type:** 32-bit integer (number of seconds) **Min value:** 15 (seconds) **Max value:** 86400 (24 hours). **Additional values accepted:** 0 is accepted and means "same as default" **Exception:** Beyond the maximum value above, the special value of 432000 (5 days) is also accepted and honored. Setting a value outside the min-max range, except if value is 0 or 432000, will result in an error.
expiry_text	String	N	If specified and the recipient's device platform supports it, the message is replaced with this text content if it exceeds the TTL before being delivered. **Default if unspecified:** 'This message has expired.' Currently supported on iOS devices only.

***Note on Media content combinations**

Different IM channels may support different combinations of text, images and actions. Below are some suggested media content combinates but IM channels may support different combinations.

- Text only
- Image only
- Text + Image + Action
- Text + Action

```
{
   . . .
   "im_content":[{
         "channel":"MESSENGER",
         "sender_id": "CompanyABC",
         "content":"This is a test message.",
         "images": [
             {
                 "url": "http://example.com"
             }
         ],
         "actions": [
             {
                 "title": "Renew now!",
                 "target_url": "http://example.com",
             }
         ],
         "ttl": 3600,
         "expiry_text": "Sorry, this promo has expired!"
      }],
   . . .
}
```

Ubiquity Text (TEXT)

Key	Type	Required	Description
text	String	Y	String containing the actual message text to send.

```
{
   . . .
    "message": {
      . . .
        "texts": [{
            "text": "This is a test message"
        }],
       . . .
    }
    . . .
}
```

Ubiquity IM Image (IM_IMAGE)

Key	Type	Required	Description
url	String	N	The HTTP URL of the image to send with the IM message. Images must be hosted on a server accessible from the Internet. One may Requiredly specify keys within the image URL to replace later with values given by the params field in the DESTINATION_INFO object. That way, one can display a different image to each message recipient. See 'params' in the DESTINATION_INFO section for more information on supplying the value for these keys. Each placeholder must be specified as {#KEY}, where KEY is a key name in the params list.

```
{
   . . .
    "message": {
      . . .
        "images": [{
            "url": "http://example.com/fiftypercent.jpg"
        }],
       . . .
    }
    . . .
}
```

Ubiquity IM Action (IM_ACTION)

Key	Type	Required	Description
title	String	Y	The caption text to display on the button.
target_url	String	Y	The HTTP URL that will be called on the click event for this action. The destination URL must be accessible from the Internet. One may Requiredly specify keys within the URL to replace later with values given by the params field in the DESTINATION_INFO object. That way, one could customise the URL depending on the message recipient opening it. See 'params' in the DESTINATION_INFO section for more information on supplying the value for these keys. Each placeholder must be specified as {#KEY}, where KEY is a key name in the params list.

```
{
   . . .
    "message": {
      . . .
        "actions": [{
            "title": "Renew now!",
            "target_url": "http://example.com/example.php"
        }]
       . . .
    }
    . . .
}
```

4.1.2. Response

Outer Response Block (OUTER_RESPONSE)

Key	Type	Required	Description
api_job_id	String	Y	A UUID identifying the life-cycle of this individual request / response, along with any future pending tasks relating to this request. Useful for error investigations and chaining.
client_job_id	String	N	If applicable, the job_id you supplied in the request is echoed here.
results	Key-Value	N	If applicable, this array contains results for each destination (see RESULT Object).
result_info	RESULT_INFO	Y	Contains a detailed description of the HTTP status code.

```
{
   "api_job_id": "763dd180-6fdd-4991-b433-a4f731ee7db1",
   "client_job_id": "abc123456",
   "result_info": {
        . . .
   },
   "results": {
         . . .
   }
}
```

Result Info Object (RESULT_INFO)

Key	Type	Required	Description
status_code	Int	Y	Same as HTTP status code.
Description	String	Y	Detailed information about status indicated.

```
{
   "api_job_id": "763dd180-6fdd-4991-b433-a4f731ee7db1",
   "client_job_id": "abc123456",
   "result_info": {
        . . .
   },
   "results": {
         . . .
   }
}
```

RESULT Object (RESULT)

Key	Type	Required	Description
{destination_msisdn}	RESULT_DETAIL	Y	Each RESULT object includes key (destination MSISDN) and value (its details), See RESULT_DETAIL Object for more info.

```
{
   "api_job_id": "763dd180-6fdd-4991-b433-a4f731ee7db1",
   "client_job_id": "abc123456",
   "result_info": {
         . . .
   },
   "results": {
         "35688000000": {
           . . .
         }
   }
}
```

Result Detail (RESULT_DETAIL)

Key	Type	Required	Description
message_id	String	Y	If the request was successful, a unique ID generated by the API for the message to this particular destination is returned here.
custom_id	String	N	If supplied, the custom_id supplied by the client from the request is echoed here.

```
{
   "api_job_id": "763dd180-6fdd-4991-b433-a4f731ee7db1",
   "client_job_id": "abc123456",
   "result_info": {
         . . .
   },
   "results": {
         "35688000000": {
            "message_id": "14474342341620014003",
            "custom_id": "1"
         }
   }
}
```

4.1.3. Example

Send SMS

HTTP

```
POST /1/im HTTP/1.1
Host: rest.fortytwo.com
Authorization: Token 424acc5f-4dbf-4c09-a81c-bb256XXXXXX
Content-Type: application/json; charset=utf-8
Cache-Control: no-cache
{
    "destinations": [{
        "number": "35688000000"
    }],
    "sms_content": {
        "sender_id":"Sales ABC",
        "message": "50% discount on our membership fee if you renew your
subscription today!"
    }
}
```

PHP

```
$curl = curl_init();

curl_setopt_array($curl, array(
  CURLOPT_URL => "https://rest.fortytwo.com/1/im",
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_ENCODING => "",
  CURLOPT_MAXREDIRS => 10,
  CURLOPT_TIMEOUT => 30,
  CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
  CURLOPT_CUSTOMREQUEST => "POST",
   CURLOPT_POSTFIELDS => "{\n     \"destinations\": [\n     {\n          \"number\":
\"35688000000\"\n    }\n    ],\n    \"sms_content\": {\n       \"sender_id\":\"Sales
ABC\",\n        \"message\": \"This is a test message\"\n    }\n}",
  CURLOPT_HTTPHEADER => array(
    "authorization: Token e927d11b-ab76-4400-b008-XXXXXX",
    "content-type: application/json; charset=utf-8"
  ),
));

$response = curl_exec($curl);
$err = curl_error($curl);

curl_close($curl);

if ($err) {
  echo "cURL Error #:" . $err;
} else {
  echo $response;
}
```

Python

```
import requests

url = "https://rest.fortytwo.com/1/im"

payload = "{\n    \"destinations\": [\n    {\n        \"number\": \"35688000000\"\n
}\n      ],\n      \"sms_content\":  {\n            \"sender_id\":\"Sales  ABC\",\n
\"message\": \"This is a test message\"\n    }\n}"
headers = {
    'content-type': "application/json; charset=utf-8",
    'authorization': "Token e927d11b-ab76-4400-b008-XXXXXX"
    }

response = requests.request("POST", url, data=payload, headers=headers)

print(response.text)
```

Send SMS to Multiple Destinations

HTTP

```
POST /1/im HTTP/1.1
Host: https://rest.fortytwo.com
Content-Type: application/json; charset=utf-8
Authorization: Token e927d11b-ab76-4400-b008-XXXXXX
{
    "destinations": [
    {
        "number": "35688000000"
    },
    {
        "number": "35688000001"
    }
    ],
    "sms_content": {
        "sender_id":"Sales ABC",
        "message": "This is a test message"
    }
}
```

HTTP

```
$curl = curl_init();

curl_setopt_array($curl, array(
  CURLOPT_URL => "https://rest.fortytwo.com/1/im",
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_ENCODING => "",
  CURLOPT_MAXREDIRS => 10,
  CURLOPT_TIMEOUT => 30,
  CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
  CURLOPT_CUSTOMREQUEST => "POST",
   CURLOPT_POSTFIELDS => "{\n     \"destinations\": [\n     {\n          \"number\":
\"35688000000\"\n    },\n    {\n        \"number\": \"35688000001\"\n    }\n    ],\n
\"sms_content\": {\n        \"sender_id\":\"Sales ABC\",\n       \"message\": \"This
is a test message\"\n    }\n}",
  CURLOPT_HTTPHEADER => array(
    "authorization: Token e927d11b-ab76-4400-b008-XXXXXX",
    "content-type: application/json; charset=utf-8"
  ),
));

$response = curl_exec($curl);
$err = curl_error($curl);

curl_close($curl);

if ($err) {
  echo "cURL Error #:" . $err;
} else {
  echo $response;
}
```

PYTHON

```
import requests

url = "https://rest.fortytwo.com/1/im"

payload = "{\n    \"destinations\": [\n    {\n        \"number\": \"35688000000\"\n
},\n    {\n        \"number\": \"35688000001\"\n    }\n    ],\n    \"sms_content\":
{\n          \"sender_id\":\"Sales ABC\",\n          \"message\": \"This is a test
message\"\n    }\n}"
headers = {
    'content-type': "application/json; charset=utf-8",
    'authorization': "Token e927d11b-ab76-4400-b008-XXXXXX"
    }

response = requests.request("POST", url, data=payload, headers=headers)

print(response.text)
```

Send Ubiquity Message to IM & SMS

HTTP

```
POST /1/im HTTP/1.1
Host: https://rest.fortytwo.com
Content-Type: application/json; charset=utf-8
Authorization: Token e927d11b-ab76-4400-b008-XXXXXX
{
    "destinations": [{
        "number": "35688000000"
    }],
    "message": {
        "texts": [{
            "text": "This is a test message"
        }]
    }
}
```

PHP

```
$curl = curl_init();

curl_setopt_array($curl, array(
  CURLOPT_URL => "https://rest.fortytwo.com/1/im",
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_ENCODING => "",
  CURLOPT_MAXREDIRS => 10,
  CURLOPT_TIMEOUT => 30,
  CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
  CURLOPT_CUSTOMREQUEST => "POST",
 CURLOPT_POSTFIELDS => "{\n  \"destinations\": [{\n    \"number\": \"35688000000\"\n
}],\n    \"message\": {\n        \"texts\": [{\n            \"text\": \"This is a
test message\"\n        }]\n    }\n}",
  CURLOPT_HTTPHEADER => array(
    "authorization: Token e927d11b-ab76-4400-b008-XXXXXX",
    "content-type: application/json; charset=utf-8"
  ),
));

$response = curl_exec($curl);
$err = curl_error($curl);

curl_close($curl);

if ($err) {
  echo "cURL Error #:" . $err;
} else {
  echo $response;
}
```

PYTHON

```
import requests

url = "https://rest.fortytwo.com/1/im"

payload = "{\n      \"destinations\": [{\n          \"number\": \"35688000000\"\n
}],\n    \"message\": {\n        \"texts\": [{\n            \"text\": \"This is a
test message\"\n        }]\n    }\n}"
headers = {
    'content-type': "application/json; charset=utf-8",
    'authorization': "Token e927d11b-ab76-4400-b008-XXXXXX"
    }

response = requests.request("POST", url, data=payload, headers=headers)

print(response.text)
```

Send Instant Message

HTTP

```
POST /1/im HTTP/1.1
Host: https://rest.fortytwo.com
Content-Type: application/json; charset=utf-8
Authorization: Token e927d11b-ab76-4400-b008-XXXXXX
{
  "destinations": [{
    "number": "35688000000"
  }],
  "im_content": [{
    "content": "This is a test message"
  }]
}
```

PHP

```php
$curl = curl_init();
curl_setopt_array($curl, array(
  CURLOPT_URL => "https://rest.fortytwo.com/1/im",
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_ENCODING => "",
  CURLOPT_MAXREDIRS => 10,
  CURLOPT_TIMEOUT => 30,
  CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
  CURLOPT_CUSTOMREQUEST => "POST",
  CURLOPT_POSTFIELDS => "{\n  \"destinations\": [{\n    \"number\": \"35688000000\"\n
}],\n  \"im_content\": [{\n    \"content\": \"This is a test message\"\n  }]\n}",
  CURLOPT_HTTPHEADER => array(
    "authorization: Token e927d11b-ab76-4400-b008-XXXXXX",
    "content-type: application/json; charset=utf-8"
  ),
));
$response = curl_exec($curl);
$err = curl_error($curl);
curl_close($curl);
if ($err) {
  echo "cURL Error #:" . $err;
} else {
  echo $response;
}
```

PYTHON

```python
import requests

url = "https://rest.fortytwo.com/1/im"

payload = "{\n  \"destinations\": [{\n    \"number\": \"35688000000\"\n  }],\n  \"im_
content\": [{\n    \"content\": \"This is a test message\"\n  }]\n}"
headers = {
    'content-type': "application/json; charset=utf-8",
    'authorization': "Token e927d11b-ab76-4400-b008-XXXXXX"
    }

response = requests.request("POST", url, data=payload, headers=headers)

print(response.text)
```

Send Instant Message with Text, Image and Button

HTTP

```
POST /1/im HTTP/1.1
Host: https://rest.fortytwo.com
Content-Type: application/json; charset=utf-8
Authorization: Token e927d11b-ab76-4400-b008-XXXXXX
{
   "destinations":[
      {
         "number":"35688000000"
      }
   ],
   "im_content":[
      {
         "content":"This is a test message",
         "images": [
             {
                 "url": "http://example.com/image.jpg"
             }
         ],
         "actions": [
             {
                 "title": "Click Me",
                 "target_url": "http://example.com/offer"
             }
         ]
      }
   ]
}
```

PHP

```php
$curl = curl_init();

curl_setopt_array($curl, array(
  CURLOPT_URL => "https://rest.fortytwo.com/1/im",
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_ENCODING => "",
  CURLOPT_MAXREDIRS => 10,
  CURLOPT_TIMEOUT => 30,
  CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
  CURLOPT_CUSTOMREQUEST => "POST",
 CURLOPT_POSTFIELDS => "{\n \"destinations\":[\n  {\n   \"number\":\"35688000000\"\n
}\n    ],\n    \"im_content\":[\n        {\n           \"content\":\"This is a test
message\",\n        \"images\": [\n            {\n                \"url\": \"http://
example.com/image.jpg\"\n              }\n          ],\n          \"actions\": [\n
{\n                  \"title\": \"Click Me\", \n                   \"target_url\":
\"http://example.com/offer\"\n             }\n         ]\n      }\n   ]\n}",
  CURLOPT_HTTPHEADER => array(
    "authorization: Token e927d11b-ab76-4400-b008-XXXXXX",
    "content-type: application/json; charset=utf-8"
  ),
));

$response = curl_exec($curl);
$err = curl_error($curl);

curl_close($curl);

if ($err) {
  echo "cURL Error #:" . $err;
} else {
  echo $response;
}
```

PYTHON

```python
import requests

url = "https://rest.fortytwo.com/1/im"

payload = "{\n   \"destinations\":[\n      {\n         \"number\":\"35688000000\"\n
}\n    ],\n    \"im_content\":[\n        {\n           \"content\":\"This is a test
message\",\n        \"images\": [\n            {\n                \"url\": \"http://
example.com/image.jpg\"\n              }\n          ],\n          \"actions\": [\n
{\n                  \"title\": \"Click Me\", \n                   \"target_url\":
\"http://example.com/offer\"\n             }\n         ]\n      }\n   ]\n}"
headers = {
    'content-type': "application/json; charset=utf-8",
    'authorization': "Token e927d11b-ab76-4400-b008-XXXXXX"
    }

response = requests.request("POST", url, data=payload, headers=headers)

print(response.text)
```

Send Instant Message with SMS Fallback

HTTP

```
POST /1/im HTTP/1.1
Host: https://rest.fortytwo.com
Content-Type: application/json; charset=utf-8
Authorization: Token e927d11b-ab76-4400-b008-XXXXXX
{
   "destinations":[
      {
         "number":"35688000000"
      }
   ],
   "sms_content":{
        "message": "This is a test message from SMS"
   },
   "im_content":[
      {
         "content":"This is a test message from IM"
      }
   ]
}
```

PHP

```
$curl = curl_init();

curl_setopt_array($curl, array(
  CURLOPT_URL => "https://rest.fortytwo.com/1/im",
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_ENCODING => "",
  CURLOPT_MAXREDIRS => 10,
  CURLOPT_TIMEOUT => 30,
  CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
  CURLOPT_CUSTOMREQUEST => "POST",
 CURLOPT_POSTFIELDS => "{\n \"destinations\":[\n  {\n   \"number\":\"35688000000\"\n
}\n   ],\n   \"sms_content\":{\n        \"message\": \"This is a test message from
SMS\"\n   },\n   \"im_content\":[\n      {\n         \"content\":\"This is a test
message from IM\"\n      }\n   ]\n} ",
  CURLOPT_HTTPHEADER => array(
    "authorization: Token e927d11b-ab76-4400-b008-XXXXXX",
    "content-type: application/json; charset=utf-8"
  ),
));

$response = curl_exec($curl);
$err = curl_error($curl);

curl_close($curl);

if ($err) {
  echo "cURL Error #:" . $err;
} else {
  echo $response;
}
```

PYTHON

```
import requests

url = "https://rest.fortytwo.com/1/im"

payload = "{\n   \"destinations\":[\n      {\n         \"number\":\"35688000000\"\n
}\n   ],\n   \"sms_content\":{\n         \"message\": \"This is a test message from
SMS\"\n   },\n   \"im_content\":[\n      {\n         \"content\":\"This is a test
message from IM\"\n      }\n   ]\n} "
headers = {
    'content-type': "application/json; charset=utf-8",
    'authorization': "Token e927d11b-ab76-4400-b008-XXXXXX"
    }

response = requests.request("POST", url, data=payload, headers=headers)

print(response.text)
```

4.2. Check Message Status

You can poll our service to check the status of a particular message ID.

To extract information for a particular IM or SMS, send a HTTP GET request to this endpoint. It will return information such as Message Type, Status, Timestamps, Source and Destination numbers, and other meta data that might help you determine the outcome of the message.

The GET endpoints takes the Message ID as a parameter.

4.2.1. Request

`GET` `https://rest.fortytwo.com/1/im/status/{message_id}`

4.2.2. Response

```
{
  "api_job_id": "77fb34ca-55f0-440e-b432-3c6aac7d9731",
  "client_job_id": "cc86c07d-5282-4b6a-b3c0-720cea542b96",
  "data": [
    {
      "type": "MESSENGER",
      "message_id": "1489422434699002XXXX",
      "status": "SEEN",
      "timestamp": 1489422440,
      "micro_timestamp": 1489422440455,
      "to": "35688000000",
      "sms_from": "Company ABC",
      "messenger_from": "Company ABC",
      "client_message_id": "",
      "error_code": 0,
      "#meta": {}
    }
  ]
}
```

4.2.3. Example

HTTP

```
[GET] https://rest.fortytwo.com/1/im/status/14897658075970XXXXX
```

PHP

```
$curl = curl_init();

curl_setopt_array($curl, array(
  CURLOPT_URL => "https://rest.fortytwo.com/1/im/status/14897658075970XXXXX",
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_ENCODING => "",
  CURLOPT_MAXREDIRS => 10,
  CURLOPT_TIMEOUT => 30,
  CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
  CURLOPT_CUSTOMREQUEST => "GET",
  CURLOPT_POSTFIELDS => "",
  CURLOPT_HTTPHEADER => array(
    "authorization: Token 87b54f4d-5c46-4d06-a266-b0b51XXXXX",
    "cache-control: no-cache",
    "content-type: application/json; charset=utf-8"
  ),
));

$response = curl_exec($curl);
$err = curl_error($curl);

curl_close($curl);

if ($err) {
  echo "cURL Error #:" . $err;
} else {
  echo $response;
}
```

CURL

```
curl -X GET -H "Content-Type: application/json; charset=utf-8" -H "Authorization:
Token      87b54f4d-5c46-4d06-a266-b0b51XXXXX"      "https://rest.fortytwo.com/1/im/
status/14897658075970XXXXX"
```

PYTHON

```
import requests

url = "https://rest.fortytwo.com/1/im/status/14897658075970XXXXX"

payload = ""
headers = {
    'content-type': "application/json; charset=utf-8",
    'authorization': "Token 87b54f4d-5c46-4d06-a266-b0b51XXXXX",
    'cache-control': "no-cache"
    }

response = requests.request("GET", url, data=payload, headers=headers)

print(response.text)
```

5. CALLBACKS

AMP supports HTTP POST callbacks, meaning that you can have an endpoint on your own server which will receive callbacks whenever something happens. This is used to track delivered messages & allow you to build custom statistics. This callback URL must be publicly available on the internet and have our IP whitelisted (if the system has a firewall).

Callbacks are separated into two categories: delivery status reports and reply messages. Both are signalled back over an HTTP callback to the URLs supplied in the original message request (see Request Body » Outer Level).

Delivery Reports and any other relevant intermediary statuses are posted to the URL specified by the “callback_url” field in the original message request. If no callback_url was specified in the request, no status report callback will be generated. Since more than one delivery report may be obtained from the destination network in a very short time frame, each post request may contain multiple delivery reports bundled together.

For two-way messaging, any reply messages sent by the recipient of the original message will be posted to the URL given by the “reply_url” field in the original message request. If no reply_url is set, two-way messaging will be disabled.

5.1. Whitelist callback server

You may need to configure your firewall to whitelist traffic from these IP addresses:

IPv4: 80.252.167.60

Note that these IPs can change in the future.

5.2. SSL Support

The use of SSL on the client's callback server is optional, i.e. both HTTP and HTTPS schemes are supported. However, if HTTPS validation fails, the callback is treated as failed. For information on how failures are handled, see Callback failure below.

5.3. Callback failure

The HTTP Callback on your server should return the correct HTTP Headers containing 200 OK. If the client's server is unreachable or does not return a 200 OK, the callback is queued to be retried again. After three failed attempts, with an interval of 5 minutes between them, the callback is discarded.

5.4. Message status callback

JSON Response

Key	Type	Required	Description
title	String	Y	The caption text to display on the button.
api_job_id	String	Y	The Job ID generated by the API. This is the same ID that was returned in the response for the request sent in Request Body.
client_job_id	String	N	The Job ID supplied by yourself during the original request in Request Body, if supplied.
data	CALLBACK_INFO[]	Y	This contains always at least 1 record.

```
{
   "api_job_id":"4c2478d3-aebb-4510-8720-1b479d01cfd5",
   "client_job_id":"abc123456",
   "data":[
      {
         . . .
      }
   ]
}
```

Callback info(CALLBACK_INFO)

Key	Type	Required	Description
type	Enum	Y	The channel through which the delivery report originated from. Enum: INTERNAL — Initial phase of the request SMS — SMS Network RCS — RCS Network MESSENGER — IM Service provided by Fortytwo TELEGRAM — IM Service on Telegram Network WHATSAPP — IM Service on Whatsapp Network

Key	Type	Required	Description
message_id	String	Y	The message ID for this message. This should match the message_id returned in the response detailed in Response.
status	Enum	Y	Enumeration indicating the message's state at the moment of the call-back generation. In case of SMS reports, these match the worldwide SMPP v3.4 standard. As for IM networks, extra custom statuses were added to support these new networks.Statuses for IM extend, not replace, the SMPP standard. Currently, the following statuses can be returned:
timestamp	long	Y	UNIX timestamp (i.e. number of seconds from 1 -Jan-1970 in UTC) when the delivery was completed.
micro_timestamp	long	Y	UNIX micro-timestamp (i.e. number of milliseconds from 1 -Jan-1970 in UTC) when the delivery was completed.
to	String	Y	Destination mobile number the SMS was sent to.
from	String	Y	The SMS or IM Sender ID used to send the message.
client_message_id	String	N	The ID originally provided by yourself.
error_code	short	Y	Error code if an error occurred, or 0 if successful. For a full list of these: https://s3-eu-west-1.amazonaws.com/www.fortytwo.com/assets/Fortytwo_Telecom_error_codes.pdf

Enum	Description
ACCEPTD	Message has been accepted.
DELIVRD	Message is delivered to destination.
EXPIRED	Message validity period has expired.
DELETED	Message has been deleted.
UNDELIV	Message undeliverable.
REJECTD	Message rejected.
UNKNOWN	Message is in an invalid state.
SEEN	Message seen by the user on device

```
{
  . . .
   "data":[
      {
         "type":"MESSENGER",
         "message_id":"14182390945378 44XXX",
         "status":"SEEN",
         "timestamp":1422885283,
         "micro_timestamp":1422885283477,
         "to":"35688000000",
         "from":"CompanyA",
         "client_message_id":"2",
         "error_code":0
      }
   ]
   . . .
}
```

5.5. Message reply callback

JSON Response

Key	Type	Required	Description
api_job_id	String	Y	The api_job_id of the original message to which this reply pertains.
client_job_id	String	N	The client_job_id of the original message to which this reply pertains, if applicable.
reply_message_id	String	Y	A unique ID assigned to this reply message. Note that while req_uuid is unique for every callback, including retries, a reply_message_id is unique to the message itself.
orig_message_id	String	N	The message_id of the original message to which this reply pertains.
req_uuid	String	Y	A UUID generated by Fortytwo's systems as a unique identifier of every callback made, for traceability and troubleshooting purposes.
channel	Enum	Y	The channel on which the reply originated from. **Enum** / **Description** RCS — **Reply was sent by a user via the RCS network.** MESSENGER — **Reply was sent by a user via Fortytwo's Broadcast platform.** TELEGRAM — **Reply was sent by a Telegram user via the Telegram platform.** WHATSAPP — **Reply was sent by a Whatsapp user via the Whatsapp platform.**
timestamp	long	Y	UNIX timestamp (i.e. number of seconds from 1-Jan-1970 in UTC) when the delivery was completed.
micro_timestamp	long	Y	UNIX micro-timestamp (i.e. number of milliseconds from 1-Jan-1970 in UTC) when the delivery was completed.
from	String	Y	The Sender ID of the client replying to the original message. This is typically the client's mobile number.
to	String	Y	The Sender ID to which this reply is being sent. This is equivalent to the Sender ID used to send the original message.
message	REPLY_MSG	Y	The content of the reply. See REPLY_MSG.

```
{
    "req_uuid": "209b9f52-10a7-4d69-ae53-67ae8f237d1e",
      "api_job_id": "d4f9b368-7812-4f21-903a-8b5325d87f6a",
      "client_job_id": "test_job_id",
      "channel": "MESSENGER",
      "reply_message_id": "1489740493768OOXXXXX",
      "orig_message_id": "148974044535800XXXXX",
      "timestamp": 1458650706,
      "micro_timestamp": 1458650706000,
      "from": "35688000000",
      "to": "Fortytwo",
      "message": {
            "text": "Hello, this is a reply"
      }
}
```

Reply Message (REPLY_MSG)

Key	Type	Required	Description
text	String	N	If the reply message body contains text, this field carries the text sent by the client.

```
    "req_uuid": "209b9f52-10a7-4d69-ae53-67ae8f237d1e",
      "api_job_id": "d4f9b368-7812-4f21-903a-8b5325d87f6a",
      "client_job_id": "test_job_id",
      "channel": "MESSENGER",
      "reply_message_id": "1489740493768OOXXXXX",
      "orig_message_id": "148974044535800XXXXX",
      "timestamp": 1458650706,
      "micro_timestamp": 1458650706000,
      "from": "35688000000",
      "to": "Fortytwo",
      "message": {
            "text": "Hello, this is a reply"
      }
}
```

5.6.Server Example

Examples for Servers that accept HTTP POST Callbacks

PHP

```
$postRawData = file_get_contents("php://input");
$json = json_decode($postRawData, true);
var_dump($json);
```

6. REFERENCES

6.1. HTTP Status Code

When responding, the REST API will make use of the appropriate and relevant HTTP status code to describe the nature of the result.

The response codes are mapped as follows:

HTTP Status	Description
200	All OK.
400	Bad Request – The request is invalid and was not understood by the API.
401	Unauthorized – Header "Authorization" missing, invalid, or revoked; and / or, your host IP is not in the authorized IPs list.
403	The request contains invalid or illegal values.
404	Not Found – The endpoint on which the request was sent to, does not exist, or does not implement the API requested.
405	Method Not Allowed – If the endpoint received a request using an HTTP method (ex. GET instead of POST) that is not allowed by that endpoint.
413	Request entity was too large.
415	Unsupported Media Type – If the request was in a content-type not supported by the endpoint (e.g. text/plain instead of application/json).
429	Too many requests (Throttling).
500	Internal Server Error.

THANK YOU

Send us a message on
support@fortytwo.com
sales@fortytwo.com

42 Telecom Ltd.

fortytwo