

# Channel API Guide

Product User Guide  
--confidential--



## About ValueFirst Channel API

ValueFirst Channel API allows sending SMS as well as WhatsApp content to your subscribers using a single endpoint. WhatsApp traffic for Push notifications can be enabled for some specific templates, provided these templates have been approved by WhatsApp. ValueFirst will manage template configuration and verification on your behalf. ValueFirst Channel API provides an XML/JSON standards-based API for integrating WhatsApp capabilities into any application or an enterprise system. ValueFirst Channel API is a store and forward mechanism for sending and receiving SMS, WhatsApp messages through the API endpoint(s) to the clients.

ValueFirst Channel API is specially designed for sending Push notifications on WhatsApp through server-to-server communication. This API is available in **HTTPS** post format for both XML as well as JSON. ValueFirst Channel API provides single authentication for multiple messages and target numbers. The endpoints for this API are based on Message Queue architecture that provides high message throughput.

## Guidelines for Sending WhatsApp Messages

The following guidelines must be followed while using ValueFirst Channel API for sending the WhatsApp messages.

### Sender Phone Number (FROM)

- For national and international messaging, the Business WhatsApp number should be prefixed with the appropriate country code e.g. 91 in case of an Indian number.
- No special character like "-", "(", ")" or anything similar is allowed in the phone number, e.g., 91-9812345678 is disallowed.

### Receiver Phone Number (TO)

- For national and international messaging, the mobile number should be prefixed with the appropriate country code e.g., 91 in case of an Indian number.
- No special character like "-", "(", ")" or anything similar is allowed in the phone number, e.g., 91-9812345678 is disallowed.

## WhatsApp Message Details

- Length of message text should not exceed 4096 characters.
- The maximum file size allowed for uploaded media is 64MB
- In case of unavailability of user's internet connection, messages once submitted to the API will be valid for next 30 days and will be delivered as soon as user is able to access their data network.

### Encoding Procedure

ValueFirst Server accepts all content in XML/JSON. In case of XML packet, special characters in message text need to be encoded as per XML standards. As a rule of thumb all string data should be XML encoded as shown below:

**Note:** This is required only when you are submitting the requests programmatically. Web browsers automatically convert the text to HTML encoded format. The encoding for sending message through ValueFirst API consists of two steps.

#### Step 1 (FOR XML)

The following table displays the codes that have to be replaced.

Code	Replace with
#39 (single quote)	&apos;
#32 (space)	&#032;
#34 (double quote)	&quot;
>	&gt;
<	&lt;
#13 (Line feed)	&#013;
#10(form feed)	&#010;
#9(Tab)	&#009;

#### Step 2 (FOR XML)

ValueFirst API accepts all data as a form post. Therefore, it is required that all XML content needs to be URL encoded before submission on API.

Rules for encoding XML content to URL format:

1. Select for each character in messages.
2. If the ASCII value of the character is greater than 128 or smaller than 32 or the character is „\*“, „#“, „%“, „<“, „>“ or „+“, replace it with its corresponding hexadecimal (hereinafter Hex) value (2 digits with leading zero) proceeded by a „%“ character, e.g., space is encoded into %20.
  - \* is encoded into %2A
  - # is encoded into %23
  - % is encoded into %25
  - < is encoded into %3C
  - > is encoded into %3E

- + is encoded into %2B
- enter key (#13#10) is encoded into %0D%0A

## Example

### Before Encoding

```
https://api.myvfirst.com/psms/servlet/psms.Eservice2?data=<?xml version="1.0"
encoding="ISO-8859-1"?><!DOCTYPE MESSAGE SYSTEM
"http://127.0.0.1:80/psms/dtd/messagev12.dtd"><MESSAGE><USER /><SMS UDH="0"
CODING="1" TEXT="The flight # <101> "DEL" to "BLR" is delayed and it's revised time
will be informed later. Have a nice day!" PROPERTY="0" ID="1"><ADDRESS
FROM="91XXXXXXXXXX" TO="91XXXXXXXXXX" SEQ="1" TAG="some customer end
random data" /></SMS></MESSAGE>&action=send
```

### After Encoding

```
https://api.myvfirst.com/psms/servlet/psms.Eservice2?data=%3C?xml%20version=%221.
0%22%20encoding=%22ISO-8859-1%22%3E%3C!DOCTYPE%20 MESSAGE
%20SYSTEM%20%22http://127.0.0.1:80/psms/dtd/messagev12.dtd%22%20%3E%3CME
SSAGE%3E%3CUSER%3E%3CSMS%20%20UDH=%220%22%20CODING=%221%22
%20TEXT=%22The%20flight%20%23&btnG;%20&lt;101&gt;%20&quot;DEL&quot;%20to
%20&quot;BLR&quot;%20is%20delayed%20and%20it&apos;s%20%20revised%20time%
20will%20be%20informed%20later.%20Have%20a%20nice%20day!%22%20PROPERTY
=%220%22%20ID=%221%22%3E%3CADDRESS%20FROM=%22ValueFirst%22%20TO
=%2291XXXXXXXXXX%22%20SEQ=%221%22%20TAG=%22some%20clientside%20ra
ndom%20data%22%20/%3E%3C/SMS%3E%3C/MESSAGE%3E&action=send
```

## Reference Links

For XML encoding

[http://coderstoolbox.net/string/#!encoding=xml&action=encode&charset=iso\\_8859\\_1](http://coderstoolbox.net/string/#!encoding=xml&action=encode&charset=iso_8859_1)

For URL encoding

<http://meyerweb.com/eric/tools/dencoder/>

## Accessing Server Services

The username and password shall be provided by your account manager from ValueFirst. This authorisation information will be required to generate a bearer token which will be further used for availing the services of ValueFirst.

## Token Management

A bearer token is used to perform the MT actions and can be managed using service username and password. Following actions are linked to token:

1. Token Generation
2. Token Enable
3. Token Disable
4. Token Delete

Delete, enable and disable can be done for all tokens or for a specific given token as well. To perform for any of the mentioned actions for all use "Token=all".

## Token Rotation

Token generated will have expiry date which is 7 days so new token need to be generated before old token get expiry. This new token will further be used for MT actions. Please update your tokens timely to avoid any termination issues. An expiry date of the token is provided in every response of token generation request.

## Token Generation

## Sample Request

URL: <https://api.myvfirst.com/psms/api/messages/token?action=generate>

Request : POST

Authorization: Basic Auth

POST

https://api.myfirst.com/psms/api/messages/token?action=generate

Send

Params

Authorization

Headers (10)

Body

Pre-request Script

Tests

Settings

Cookies

Type

Basic Auth

The authorization header will be automatically generated when you send the request.  
[Learn more about authorization >](#)

UsernameXXXXXX

Password.....  
☐ Show Password

Body

Cookies

Headers (11)

Test Results

Status: 200 OK Time: 307 ms Size: 669 B Save Response

PrettyRawPreviewVisualizeJSON

```
1{"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6Iml0dHRwciovL2FwaSStZXZhbHVlZmlycyQyVjY29tL3BxbXNlcjZwdWIlOjI0KzW1vc291bXlhcg1sMyIsImV4cCI6MTYxOT04MDg3MzB0LnPvLBI0ZkUsbn3GHThozCn-Nl9MMCHLFo9gLjd66k",
2
3"expiryDate": "2021-04-21 17:57:53"}
4
```



## Sample Response

```
{
  "token": "eyJhbKJKJKVibskui1NiJ9.eyJpcBXMiyOBodHRwcZovL2FwaS5teXZhbHVlZmlyc3QuY29tL3BzbXMiLCJzdWIiOiJkZW1vc291bXlheG1sMyIsImV4cCI6MTYxOTE4MDg3M30.rPuvLbiNoZKusW3GHThoZCW-Nl9MMCHWLfo9gLjd66k",
  "expiryDate": "2021-04-21 17:57:53"
}
```

## Token Enable

### Sample Request (Single Token Enable)

URL: <https://api.myvfirst.com/psms/api/messages/token?action=enable&token=eyJhbKJKJKVibskui1NiJ9.eyJpcBXMiyOBodHRwcZovL2FwaS5teXZhbHVlZmlyc3QuY29tL3BzbXMiLCJzdWIiOiJkZW1vc291bXlheG1sMyIsImV4cCI6MTYxOTE4MDg3M30.rPuvLbiNoZKusW3GHThoZCW-Nl9MMCHWLfo9gLjd66k>

Request: POST

Authorization: Basic Auth

### Sample Request (All Token Enable)

URL: <https://api.myvfirst.com/psms/api/messages/token?action=enable&token=All>

Request: POST

Authorization: Basic Auth

## Token Disable

### Sample Request (Single Token Disable)

URL: <https://api.myvfirst.com/psms/api/messages/token?action=disable&token=eyJhbKJKJKVibskui1NiJ9.eyJpcBXMiyOBodHRwcZovL2FwaS5teXZhbHVlZmlyc3QuY29tL3BzbXMiLCJzdWIiOiJkZW1vc291bXlheG1sMyIsImV4cCI6MTYxOTE4MDg3M30.rPuvLbiNoZKusW3GHThoZCW-Nl9MMCHWLfo9gLjd66k>

Request: POST

Authorization: Basic Auth

### Sample Request (All Token Disable)

URL: <https://api.myvfirst.com/psms/api/messages/token?action=disable&token=All>

Request: POST

Authorization: Basic Auth

## Token Delete

### Sample Request (Single Token Delete)

URL: <https://api.myvfirst.com/psms/api/messages/token?action=delete&token=eyJhbKJKJKVibskui1NiJ9.eyJpcBXMiyOBodHRwcZovL2FwaS5teXZhbHVlZmlyc3QuY29tL3BzbXMiLCJzdWIiOiJkZW1vc291bXlheG1sMyIsImV4cCI6MTYxOTE4MDg3M30.rPuvLbiNoZKusW3GHThoZCW-Nl9MMCHWLfo9gLjd66k>

Request: POST

Authorization: Basic Auth

### Sample Request (All Token Delete)

URL: <https://api.myvfirst.com/psms/api/messages/token?action=delete&token=All>

Request: POST

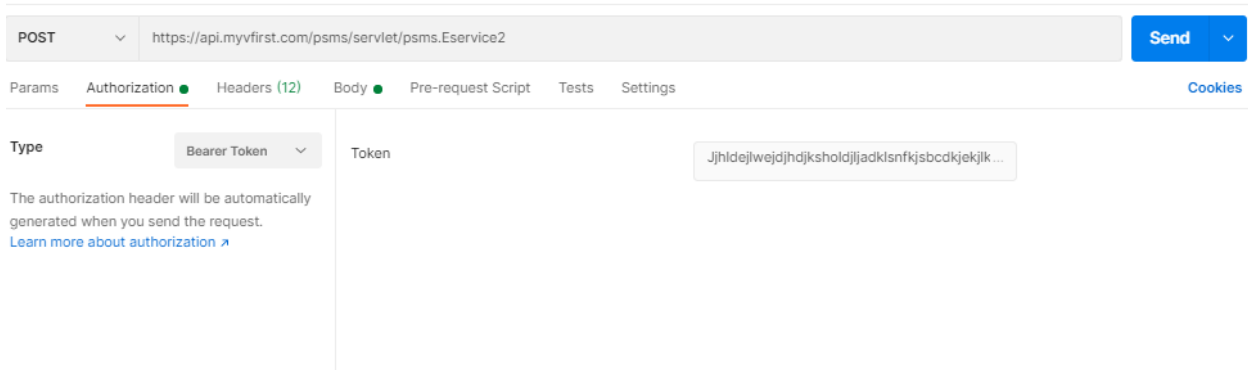
Authorization: Basic Auth

The end point for accessing ValueFirst XML API is  
<https://api.myvfirst.com/psms/servlet/psms.Eservice2>

Request: POST

Authorization: Bearer Token

Body: Raw (Content Type: Application/XML) or x-www-form-urlencoded

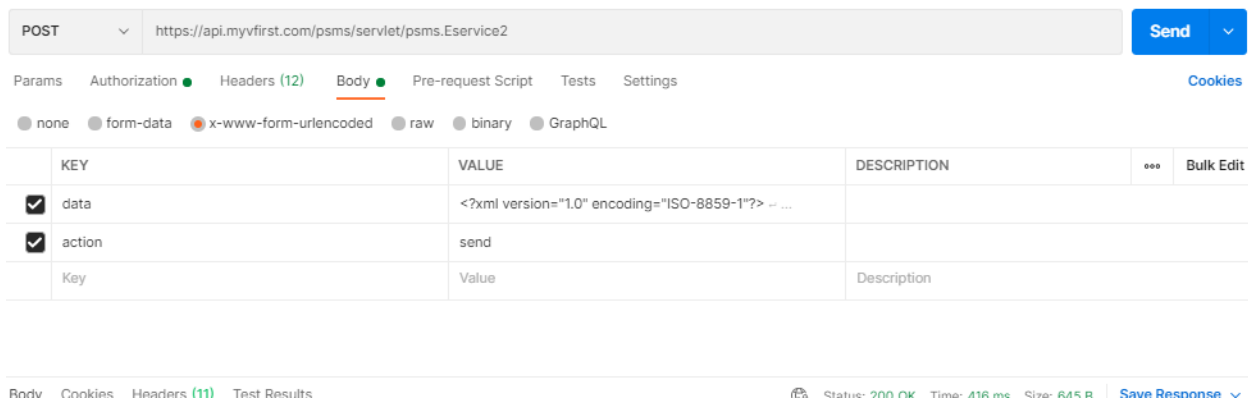


POST ▼ <https://api.myvfirst.com/psms/servlet/psms.Eservice2> Send ▼

Params Authorization Headers (12) Body Pre-request Script Tests Settings Cookies

Type Bearer Token ▼ Token Jjhldeljwejdjhdksholdjldklsnfkjsbcdkjkjlk...

The authorization header will be automatically generated when you send the request.  
[Learn more about authorization](#)



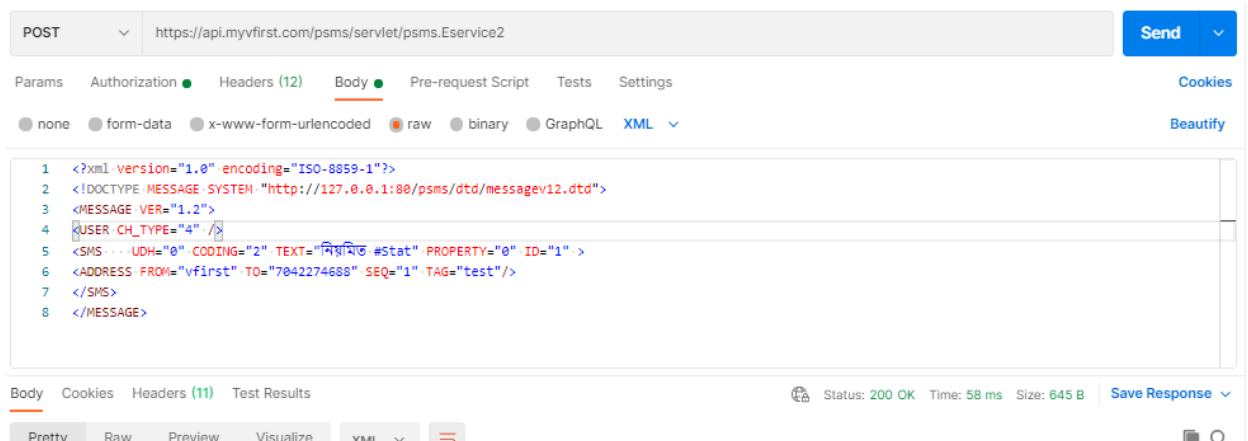
POST ▼ <https://api.myvfirst.com/psms/servlet/psms.Eservice2> Send ▼

Params Authorization Headers (12) Body Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	data	<?xml version="1.0" encoding="ISO-8859-1"?> ...			
<input checked="" type="checkbox"/>	action	send			
	Key	Value	Description		

Raw Cookies Headers (11) Test Results Status: 200 OK Time: 116 ms Size: 645 B Save Response ▼



POST ▼ <https://api.myvfirst.com/psms/servlet/psms.Eservice2> Send ▼

Params Authorization Headers (12) Body Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL XML ▼ Beautify

```

1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <!DOCTYPE MESSAGE SYSTEM "http://127.0.0.1:80/psms/dtd/messagev12.dtd">
3 <MESSAGE VER="1.2">
4   <USER_CH_TYPE="4"/>
5   <SMS ... UDH="0" CODING="2" TEXT="رسالة #Stat" PROPERTY="0" ID="1" >
6   <ADDRESS FROM="vfirst" TO="7042274688" SEQ="1" TAG="test"/>
7 </SMS>
8 </MESSAGE>

```

Body Cookies Headers (11) Test Results Status: 200 OK Time: 58 ms Size: 645 B Save Response ▼

Pretty Raw Preview Visualize XML ▼ 🔍

The end point for accessing ValueFirst JSON API is  
<https://api.myvfirst.com/psms/servlet/psms.JsonEservice>

Request: POST

Authorization: Bearer Token

Body: Raw (Content Type: Application/XML) or x-www-form-urlencoded

The above URL accepts data through three parameters namely “data”, “whatsapp\_data” and “action”. The data parameter specifies XML content that needs to be posted. The



whatsapp\_data parameter is used to send WhatsApp Request data as JSON request in some scenarios. The action parameter is different for each XML (Table 2).

Action	Data Parameter	Action Parameter
Sending message	WhatsApp Message XML/JSON (URN and HTML encoded)	send
Checking status	WhatsApp Status Request XML/JSON (URN and HTML encoded)	status
Checking credits	WhatsApp Credit Request XML/JSON (URN and HTML encoded)	credits

Note: As of now “whatsapp\_data” parameter is mandatory only in case of sending location and business card in WhatsApp two way messages.

### WhatsApp Message Sending Service

ValueFirst WhatsApp API encompasses advance features specifically designed for sending one to one as well as bulk WhatsApp messages. With these APIs, there is no need to send each message separately; multiple messages can be sent together in a single transaction. The API also supports multiple target numbers for a common message to the recipients.

### Example of Sending WhatsApp Text Push Messages (Using XML API)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE MESSAGE SYSTEM "http://127.0.0.1:80/psms/dtd/messagev12.dtd" >
<MESSAGE>
  <USER CH_TYPE="4" />
  <SMS UDH="0" CODING="1" TEXT="hey this is a real test" MSGTYPE="1"
PROPERTY="" ID="1">
    <ADDRESS FROM=" 91XXXXXXXXXX " TO="919812345678" SEQ="1"
TAG=" " />
    <ADDRESS FROM=" 91XXXXXXXXXX " TO="mycompany" SEQ="2" />
    <ADDRESS FROM=" 91XXXXXXXXXX " TO="919812345678" SEQ="3" />
  </SMS>
  <SMS UDH="0" CODING="1" TEXT="hey this is a real test" MSGTYPE="1"
PROPERTY="" ID="2" SEND_ON="2007-10-15 20:10:10 +0530">
    <ADDRESS FROM="91XXXXXXXXXX" TO="919812345678" SEQ="1" />
    <ADDRESS FROM="91XXXXXXXXXX" TO="919812345678" SEQ="2" />
  </SMS>
</MESSAGE>
```

### Example of Sending WhatsApp Text Two Way Messages (Using XML API)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE MESSAGE SYSTEM "http://127.0.0.1:80/psms/dtd/messagev12.dtd" >
<MESSAGE>
  <USER CH_TYPE="4" />
  <SMS UDH="0" CODING="1" TEXT="hey this is a real test" MSGTYPE="2"
PROPERTY="" ID="1">
    <ADDRESS FROM=" 91XXXXXXXXXX " TO="919812345678" SEQ="1"
TAG=" " />
    <ADDRESS FROM=" 91XXXXXXXXXX " TO="mycompany" SEQ="2" />
```

```

        <ADDRESS FROM="91XXXXXXXXXX" TO="919812345678" SEQ="3" />
    </SMS>
    <SMS UDH="0" CODING="1" TEXT="hey this is a real test" MSGTYPE="2"
    PROPERTY="" ID="2" SEND_ON="2007-10-15 20:10:10 +0530">
        <ADDRESS FROM="91XXXXXXXXXX" TO="919812345678" SEQ="1" />
        <ADDRESS FROM="91XXXXXXXXXX" TO="919812345678" SEQ="2" />
    </SMS>
</MESSAGE>

```

### Example of Sending WhatsApp Text Push Messages (Using JSON API)

```

{
  "@VER": "1.2",
  "USER": {
    "@CH_TYPE": "4",
    "@UNIXTIMESTAMP": ""
  },
  "SMS": [
    {
      "@UDH": "0",
      "@CODING": "1",
      "@TEXT": "Test Message",
      "@PROPERTY": "0",
      "@MSGTYPE": "1",
      "@ID": "1",
      "ADDRESS": [
        {
          "@FROM": "91XXXXXXXXXX",
          "@TO": "91XXXXXXXXXX",
          "@SEQ": "1",
          "@TAG": "some client side random data"
        }
      ]
    }
  ]
}

```

### Example of Sending WhatsApp Text Two Way Messages (Using JSON API)

```

{
  "@VER": "1.2",
  "USER": {
    "@CH_TYPE": "4",
    "@UNIXTIMESTAMP": ""
  },
  "SMS": [
    {

```

```

"@UDH": "0",
"@CODING": "1",
"@TEXT": "Test Message",
"@PROPERTY": "0",
"@MSGTYPE": "2",
"@ID": "1",
"ADDRESS": [
  {
    "@FROM": "91XXXXXXXXXXXX",
    "@TO": "91XXXXXXXXXXXX",
    "@SEQ": "1",
    "@TAG": "some client side random data"
  }
]
}
]
}

```

### Example of Sending WhatsApp Text Push Messages (Using Template ID in XML API)

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE MESSAGE SYSTEM
  "http://127.0.0.1:58080/psms/dtd/messagev12.dtd">
<MESSAGE VER="1.2">
<USER CH_TYPE="4 " />
<SMS UDH="0" CODING="1" TEXT=" " MSGTYPE=" 1 "
  TEMPLATEINFO="188181~v1~v2" PROPERTY="0" ID="1" TEMPLATE=""
  EMAILTEXT="" ATTACHMENT="">
<ADDRESS FROM="91XXXXXXXXXXXX" TO="91XXXXXXXXXXXX" EMAIL="" SEQ="1"
  TAG="some clientside random data"/>
</SMS>
</MESSAGE>

```

### Example of Sending WhatsApp Text Push Messages (Using Template ID in JSON API)

```

{
  "@VER": "1.2",
  "USER": {
    "@CH_TYPE": "4",
    "@UNIXTIMESTAMP": ""
  },
  "SMS": [
    {
      "@UDH": "0",
      "@CODING": "1",
      "@TEXT": "",

```

```

"@TEMPLATEINFO": "19898~joh~1123~2",
"@PROPERTY": "0",
"@MSGTYPE": "1",
"@ID": "2rttu",
"ADDRESS": [
  {
    "@FROM": "91XXXXXXXXXX",
    "@TO": "91XXXXXXXXXX",
    "@SEQ": "1",
    "@TAG": "some clientside random data"
  }
]
}
]
}

```

### Example of Sending WhatsApp Rich Media Push Messages (Using XML API)

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE MESSAGE SYSTEM "http://127.0.0.1:8181/psms/dtd/messagev12.dtd">
<MESSAGE VER="1.2">
  <USER CH_TYPE="4" />
  <SMS UDH="0" CODING="1" TEXT="Hi mycompany, Please find status of you
  complaint." MEDIADATA="https://www.dbs.com/in/iwov-resources/pdf/forms-and-
  legal/form-15g.pdf" MSGTYPE="3" TYPE="document" PROPERTY="0" ID="1"
  TEMPLATE="" EMAILTEXT="" ATTACHMENT="">
    <ADDRESS FROM="91XXXXXXXXXX" TO="91XXXXXXXXXX" EMAIL=""
    SEQ="1" TAG="some client side random data"/>
  </SMS>

  <SMS UDH="0" CODING="1" TEXT="Hi mycompany,Please find your logo."
  MEDIADATA="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcThpV-
  RCiOWj9jxJXQKkG9CH8QsDtjOaXsT9Reyqz03ULBZf0i3ig&s" MSGTYPE="3"
  TYPE="image" PROPERTY="0" ID="1" TEMPLATE="" EMAILTEXT=""
  ATTACHMENT="">
    <ADDRESS FROM="91XXXXXXXXXX" TO="91XXXXXXXXXX" EMAIL=""
    SEQ="1" TAG="some client side random data"/>
  </SMS>

</MESSAGE>

```

### Example of Sending WhatsApp Rich Media Two Way Messages (Using XML API)

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE MESSAGE SYSTEM "http://127.0.0.1:8181/psms/dtd/messagev12.dtd">
<MESSAGE VER="1.2">

```

```

<USER CH_TYPE="4" />
  <SMS UDH="0" CODING="1" TEXT="Hi mycompany, Please find status of you
  complaint." MEDIADATA="https://www.dbs.com/in/iwov-resources/pdf/forms-and-
  legal/form-15g.pdf" MSGTYPE="4" TYPE="document" PROPERTY="0" ID="1"
  TEMPLATE="" EMAILTEXT="" ATTACHMENT="">
    <ADDRESS FROM="91XXXXXXXXXX " TO="91XXXXXXXXXX " EMAIL=""
    SEQ="1" TAG="some client side random data"/>
  </SMS>

  <SMS UDH="0" CODING="1" TEXT="Hi mycompany,Please find your logo."
  MEDIADATA="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcThpV-
  RCiOWj9jxJXQKkG9CH8QsDtjOaXsT9Reyqz03ULBZf0i3ig&";s" MSGTYPE="4"
  TYPE="image" PROPERTY="0" ID="1" TEMPLATE="" EMAILTEXT=""
  ATTACHMENT="">
    <ADDRESS FROM="91XXXXXXXXXX " TO="91XXXXXXXXXX " EMAIL=""
    SEQ="1" TAG="some client side random data"/>
  </SMS>

</MESSAGE>

```

### Example of Sending WhatsApp Rich Media Push Messages (Using Template ID in XML API)

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE MESSAGE SYSTEM "http://127.0.0.1:80/psms/dtd/messagev12.dtd">
<MESSAGE VER="1.2">
  <USER CH_TYPE="4" />
  <SMS UDH="0" CODING="1" TEXT="" TEMPLATEINFO="191896~test"
  MEDIADATA="https://s1.q4cdn.com/806093406/files/doc_downloads/test.pdf"
  MSGTYPE="3" TYPE="document" PROPERTY="0" ID="388" ATTACHMENT="">
    <ADDRESS FROM="91XXXXXXXX" TO="91XXXXXX" EMAIL="" SEQ="1"
    TAG="some client side random data"/>
  </SMS>
</MESSAGE>

```

### Example of Sending WhatsApp Rich Media Push Messages (Using Template ID in JSON API)

```

{
  "@VER": "1.2",
  "USER": {
    "@CH_TYPE": "4 ",
    "@UNIXTIMESTAMP": ""
  },
  "DLR": {

```

```

"@URL": ""
},
"SMS": [
  {
    "@UDH": "0",
    "@CODING": "1",
    "@TEXT": "test ",
    "@TEMPLATEINFO": "19898~joh~1123~2",
    "@CONTENTTYPE": "video/mp4",
    "@TYPE": "video",
    "@MSGTYPE": "3",
    "@MEDIADATA": "https://file-examples.com/wp-
content/uploads/2017/04/file_example_MP4_480_1_5MG.mp4",
    "@PROPERTY": "0",
    "@ID": "0789",
    "ADDRESS": [
      {
        "@FROM": "91XXXXXX",
        "@TO": "91XXXXXX",
        "@SEQ": "1",
        "@TAG": "some client side random data"
      }
    ]
  }
]
}

```

### Example of Sending WhatsApp Rich Media Push Messages (Using JSON API)

```

{
  "@VER": "1.2",
  "USER": {
    "@CH_TYPE": "4 ",
    "@UNIXTIMESTAMP": ""
  },
  "DLR": {
    "@URL": ""
  },
  "SMS": [
    {
      "@UDH": "0",
      "@CODING": "1",
      "@TEXT": "test ",
      "@CONTENTTYPE": "video/mp4",

```

```

"@TYPE": "video",
"@MSGTYPE": "3",
"@MEDIADATA": "https://file-examples.com/wp-
content/uploads/2017/04/file_example_MP4_480_1_5MG.mp4",
"@PROPERTY": "0",
"@ID": "0789",
"ADDRESS": [
  {
    "@FROM": "91XXXXXX",
    "@TO": "91XXXXXX",
    "@SEQ": "1",
    "@TAG": "some client side random data"
  }
]
}
]
}

```

### Example of Sending WhatsApp Sticker in Two Way Messages (Using XML API)

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE MESSAGE SYSTEM "http://127.0.0.1:8181/psms/dtd/messagev12.dtd">
<MESSAGE VER="1.2">
  <USER CH_TYPE="4" />
  <SMS UDH="0" CODING="1" TEXT="" MSGTYPE="2" TYPE="sticker"
    PROPERTY="0" ID="346d05df-3ded-4fd0-af8b-1053876782" TEMPLATE=""
    EMAILTEXT="" ATTACHMENT="">
    <ADDRESS FROM="91XXXXXXXXXX" TO="91XXXXXXXXXX" EMAIL=""
      SEQ="1" TAG="some client side random data"/>
  </SMS>
</MESSAGE>

```

### Example of Sending WhatsApp Rich Media Two Way Messages ( Using JSON API)

```

{
  "@VER": "1.2",
  "USER": {
    "@CH_TYPE": "4 ",
    "@UNIXTIMESTAMP": ""
  },
  "DLR": {
    "@URL": ""
  },
  "SMS": [
    {
      "@UDH": "0",
      "@CODING": "1",

```



```

"@TEXT": "",
"@CAPTION": "test",
"@CONTENTTYPE": "video/mp4",
"@TYPE": "video",
"@MSGTYPE": "4",
"@MEDIADATA": "https://file-examples.com/wp-
content/uploads/2017/04/file_example_MP4_480_1_5MG.mp4",
"@PROPERTY": "0",
"@ID": "0789",
"ADDRESS": [
  {
    "@FROM": "91XXXXXX",
    "@TO": "91XXXXXX",
    "@SEQ": "1",
    "@TAG": "some client side random data"
  }
]
}
]
}

```

Note: In case of session based messages caption parameter will work and in case of push notifications text is supported.

### Example of Sending WhatsApp Location in Two Way Messages [XML]

Sample value for data parameter are below:

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE MESSAGE SYSTEM "http://127.0.0.1:80/psms/dtd/messagev12.dtd">
<MESSAGE VER="1.2">
<USER CH_TYPE="4" />
<SMS UDH="0" CODING="1" TEXT="" MSGTYPE="6" PROPERTY="0" ID="1"
TEMPLATE="" EMAILTEXT="" ATTACHMENT="">
<ADDRESS FROM="91XXXXXXXXXX" TO="91XXXXXXXXXX" EMAIL="" SEQ="1"
TAG="some clientside random data"/>
</SMS>
</MESSAGE>

```

Sample value (JSON) for whatsapp\_data parameter are below:

```

{
  "type": "location",
  "recipient_type": "individual",
  "location": {
    "longitude": "28.430944",

```

```
"latitude": "77.013683",
"name": "ValueFirst Digital Media Pvt Ltd",
"address": "B-18,Infocity-1,sector-34 (gurgaon)"
}
```

Below are the postman screenshots for the reference:

WhatsApp / Location XML

POST `https://api.myvfirst.com/psms/servlet/psms.Eservice2` Send

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> data	<?xml version="1.0" encoding="ISO-8859-1"?> ...			
<input checked="" type="checkbox"/> whatsapp_data	{ ...			
<input checked="" type="checkbox"/> action	send			
Key	Value	Description		

Body Cookies Headers (11) Test Results Status: 200 OK Time: 64 ms Size: 615 B Save Response

Pretty Raw Preview Visualize XML Search

```

1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <MESSAGEACK>
3   <GUID GUID="k14ja4759686v1f4400103s:zgj8DEMOA8AS" SUBMITDATE="2021-04-19 10:47:59" ID="1"></GUID>
4 </MESSAGEACK>
  
```

Activate Windows  
Go to Settings to activate Windows.

Location Testing

GET `https://api.myvaluefirst.com/psms/servlet/psms.Eservice2` Send Save

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings Cookies Code

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> data	<?xml version="1.0" encoding="ISO-8859-1"?> ...			
<input checked="" type="checkbox"/> whatsapp_data	{ ...			
<input checked="" type="checkbox"/> action	send			
Key	Value	Description		

Body Cookies (1) Headers (10) Test Results 200 OK Time: 726ms Size: 637 B Save Response

Pretty Raw Preview Visualize XML Search

```

1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <MESSAGEACK>
3   <GUID GUID="kk580250319311f440000dpas8DEMOSOUWYA" SUBMITDATE="2020-05-08 00:25:03" ID="1"></GUID>
4 </MESSAGEACK>
  
```

Activate Windows

## Example of Sending WhatsApp Location in Two Way Messages [JSON]

Referring to the above screenshot, in JSON API will also follow the key value structure :

**data, whatsapp\_data, and action**

**Sample Value for data parameter**

```
{
  "@VER": "1.2",
  "USER": {
    "@UNIXTIMESTAMP": "",
    "@CH_TYPE": "4"
  },
  "DLR": {
    "@URL": ""
  },
  "SMS": [
    {
      "@UDH": "0",
      "@CODING": "1",
      "@TEXT": "",
      "@PROPERTY": "0",
      "@MSGTYPE": "6",
      "@ID": "998",
      "ADDRESS": [
        {
          "@FROM": "917428306034",
          "@TO": "91XXXXXXX",
          "@SEQ": "1",
          "@TAG": "some clientside random data"
        }
      ]
    }
  ]
}
```

**Sample Value for whatsapp\_data parameter**

```
{
  "type": "location",
  "recipient_type": "individual",
  "location": {
    "longitude": "28.430944",
    "latitude": "77.013683",
    "name": "ValueFirst Digital Media Pvt Ltd",
    "address": "B-18, Infocity-1, sector-34 (gurgaon)"
  }
}
```

WhatsApp / Location JSON

POST <https://api.myvaluefirst.com/psms/servlet/psms.JsonEservice> [Send](#)

Params **Authorization** Headers (10) Body Pre-request Script Tests Settings [Cookies](#)

Type **Bearer Token** Token `eyJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRw...`

The authorization header will be automatically generated when you send the request.  
[Learn more about authorization](#)

Body Cookies Headers (11) Test Results [Status: 200 OK](#) [Time: 486 ms](#) [Size: 573 B](#) [Save Response](#)

---

WhatsApp / Location JSON

POST <https://api.myvaluefirst.com/psms/servlet/psms.JsonEservice> [Send](#)

Params **Authorization** Headers (10) **Body** Pre-request Script Tests Settings [Cookies](#)

☐ none 
 ☐ form-data 
 ☒ x-www-form-urlencoded 
 ☐ raw 
 ☐ binary 
 ☐ GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	data	{ ... }			
<input checked="" type="checkbox"/>	whatsapp_data	{ ... }			
<input checked="" type="checkbox"/>	action	send			
	Key	Value	Description		

Body Cookies Headers (11) Test Results [Status: 200 OK](#) [Time: 73 ms](#) [Size: 573 B](#) [Save Response](#)

Pretty Raw Preview Visualize JSON [Copy](#)

```

1 {
2   "MESSAGEACK": {
3     "GUID": {
4       "SUBMITDATE": "2021-04-19 10:42:49",
5       "GUID": "k14ja4249526v1f4400109k2jgbdEMONABAS",
6       "ID": 93
7     }
8   }
9 }
  
```

Activate Windows

## Example of Sending WhatsApp Business Card in Two Way Messages [XML]

Sample value for data parameter is below:

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE MESSAGE SYSTEM "http://127.0.0.1:80/psms/dtd/messagev12.dtd">
<MESSAGE VER="1.2">
<USER CH_TYPE="4" />
<SMS UDH="0" CODING="1" TEXT="" MSGTYPE="5" PROPERTY="0" ID="1"
TEMPLATE="" EMAILTEXT="" ATTACHMENT="">
<ADDRESS FROM="91XXXXXXXXXX" TO="91XXXXXXXXXX" EMAIL="" SEQ="1"
TAG="some clientside random data"/>
</SMS>
</MESSAGE>
  
```

Sample value for whatsapp\_data parameter is below:

```
{ "type": "contacts",
```

```

"recipient_type": "individual",
"contacts": [
{
"addresses": [ {
"city": "sohna",
"country": "India",
"country_code": "IN",
"state": "Haryana",
"street": "sohna",
"type": "HOME",
"zip": "122103"
}
],
"birthday": "1992-07-15",
"emails": [
{
"email": "xyz@vfirst.com",
"type": "WORK"
}
],
"ims": [],
"name": {
"first_name": "New",
"formatted_name": "New Contact",
"last_name": "Contact"
},
"org": {
"company": "vfirst"
},
"phones": [
{
"phone": "91XXXXXXX",
"type": "WORK"
}
],
"urls": []
}
]
}

```

Below is the postman screenshot for the reference:

WhatsApp / Business Card XML

Save

POST <https://api.myvaluefirst.com/psms/servlet/psms.Eservice2> Send

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

	KEY	VALUE	DESCRIPTION		Bulk Edit
<input checked="" type="checkbox"/>	data	<?xml version="1.0" encoding="ISO-8859-1"?> ...			
<input checked="" type="checkbox"/>	whatsapp_data	{ ...			
<input checked="" type="checkbox"/>	action	send			
	Key	Value	Description		

Body Cookies Headers (11) Test Results

Status: 200 OK Time: 84 ms Size: 650 B Save Response

Pretty Raw Preview Visualize XML

## Example of Sending WhatsApp Business Card in Two Way Messages [JSON]

### Sample Value for data parameter

```
{
  "@VER": "1.2",
  "USER": {
    "@UNIXTIMESTAMP": "",
    "@CH_TYPE": "4"
  },
  "DLR": {
    "@URL": ""
  },
  "SMS": [
    {
      "@UDH": "0",
      "@CODING": "1",
      "@TEXT": "",
      "@PROPERTY": "0",
      "@MSGTYPE": "5",
      "@ID": "998",
      "ADDRESS": [
        {
          "@FROM": "917428306034",
          "@TO": "91XXXXXXX",
          "@SEQ": "1",
          "@TAG": "some clientside random data"
        }
      ]
    }
  ]
}
```

Sample value for whatsapp\_data parameter is below:

```
{
  "type": "contacts",
  "recipient_type": "individual",

```

```
"contacts": [  
  {  
    "addresses": [ {  
      "city": "sohna",  
      "country": "India",  
      "country_code": "IN",  
      "state": "Haryana",  
      "street": "sohna",  
      "type": "HOME",  
      "zip": "122103"  
    }  
  ],  
  "birthday": "1992-07-15",  
  "emails": [  
    {  
      "email": "xyz@vfirst.com",  
      "type": "WORK"  
    }  
  ],  
  "ims": [],  
  "name": {  
    "first_name": "New",  
    "formatted_name": "New Contact",  
    "last_name": "Contact"  
  },  
  "org": {  
    "company": "vfirst"  
  },  
  "phones": [  
    {  
      "phone": "91XXXXXXX",  
      "type": "WORK"  
    }  
  ],  
  "urls": []  
}  
]
```



WhatsApp / Business Card JSON

POST <https://api.myvaluefirst.com/psms/servlet/psms.JsonEservice> [Send](#)

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings [Cookies](#)

none form-data x-www-form-urlencoded raw binary GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	data	{ " ...			
<input checked="" type="checkbox"/>	whatsapp_data	{ "type": "contacts", " ...			
<input checked="" type="checkbox"/>	action	send			
	Key	Value	Description		

Body Cookies Headers (11) Test Results [Save Response](#)

Pretty Raw Preview Visualize JSON [Search](#)

```

1 {
2   "MESSAGEACK": {
3     "GUID": "k14ja5247113w1f4400105xbevcdEMOWABAS",
4     "SUBMITDATE": "2021-04-19 10:52:47",
5     "GUID": "k14ja5247113w1f4400105xbevcdEMOWABAS",
6     "ID": 93
7   }
8 }
9

```

Activate Windows  
Go to Settings to activate Windows.

## Sending Buttons in WhatsApp Push Notifications

Interactive message templates expand the content you can send recipients beyond the standard message template and media messages template types to include interactive buttons.

There are two types of predefined buttons offered:

**Call-to-Action** — Allows your customer to call a phone number and visit a website

**Quick Reply** — Allows your customer to return a plain text message

These buttons can be attached to text messages or media messages. Once your interactive message templates have been created and approved, you can use them in notification messages. When your customer clicks on a quick reply button, a response is sent. Below are details related to button feature:

- Button DLR is received only for quick reply buttons not for call-to-action buttons.
- Quick reply button can be clicked only once on WhatsApp as it gets disabled once clicked.
- In case of dynamic URL present in call to action its mandatory to give B\_URLINFO in the request packet as we need to pass variable in dynamic URL. Refer below packet for same.  

```

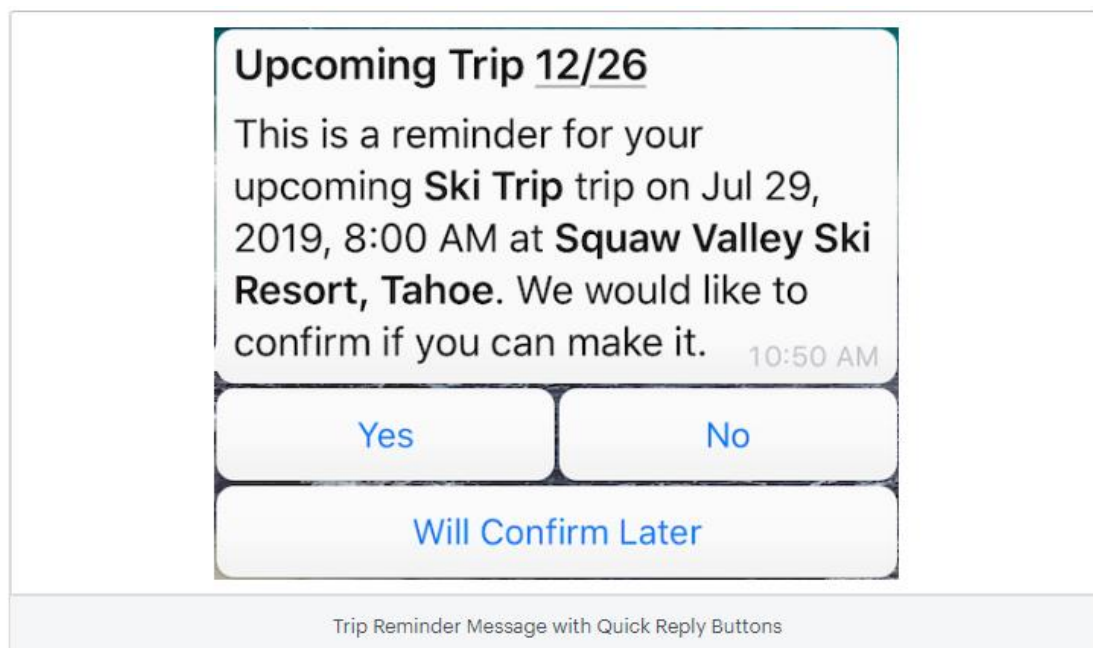
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE MESSAGE SYSTEM
"http://127.0.0.1:8181/psms/dtd/messagev12.dtd">
<MESSAGE VER="1.2">
<USER CH_TYPE="4"/>
<SMS UDH="0" CODING="1" TEXT="Thanks for contacting support."
MEDIADATA="https://winaero.com/blog/wp-content/uploads/2019/11/Photos-new-
icon.png" MSGTYPE="3" B_URLINFO="V11" TYPE="image" PROPERTY="0"
ID="1" TEMPLATE="" EMAILTEXT="" ATTACHMENT="">

```

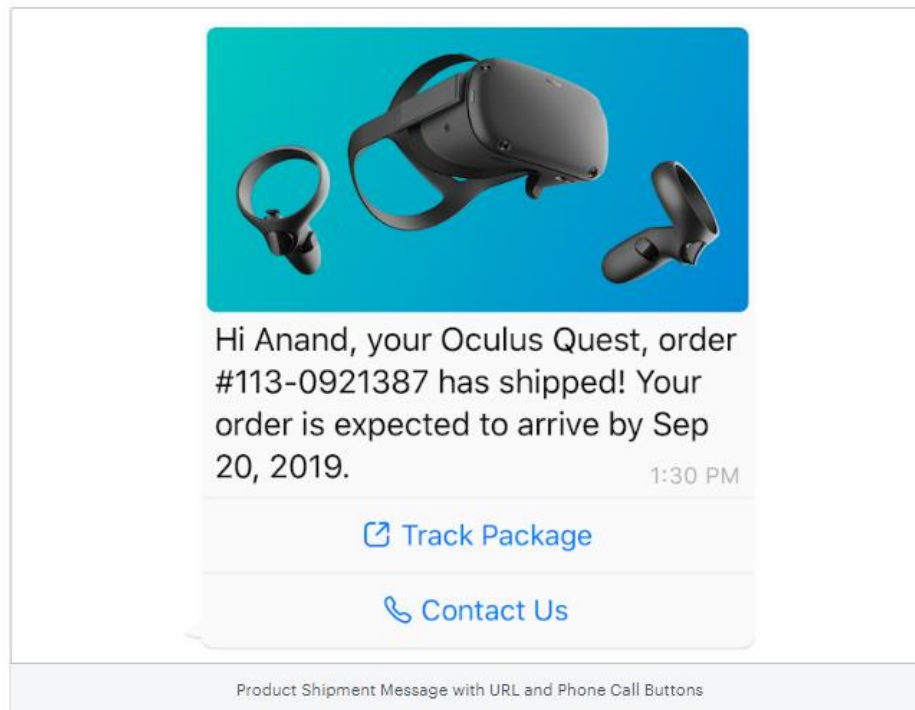
```
<ADDRESS FROM="91XXXXXXX" TO="91XXXXXXXXXX" EMAIL="" SEQ="1"  
TAG="some client side random data"/>  
</SMS>  
</MESSAGE>
```

- As of now only one variable is supported in dynamic value of URL.
- Quick reply button label max length is 20 characters only.
- All button (interactive templates) can be send using MSGTYPE="3" in request packet as its mandatory else you will receive a template related error.
- In case of quick reply maximum number of buttons can be 3.
- In case of call to action, maximum number of buttons can be 2 and only one call and one URL button can be configured as both can neither be Call or URL button.

#### EXAMPLE 1: YOUR CUSTOMER RECEIVES THEIR TRIP REMINDER MESSAGE WITH QUICK REPLY BUTTONS.



## EXAMPLE 2: YOUR CUSTOMER RECEIVES THEIR PRODUCT SHIPMENT MESSAGE WITH URL AND PHONE CALL BUTTONS.



The following table describes the different elements of a WhatsApp message request.

Tag Name	Description
<b>Users</b>	
CH_TYPE	Channel Type Value (E.g., CH_TYPE=4 for WhatsApp)
<b>WhatsApp</b>	
ID	Unique ID of message. This value is generated and sent by client application. While sending the delivery reports, server sends this value back to the client application. This value can also be used check status of the message.
TEXT	Content of Message/Caption to be sent with media message.
MSGTYPE	Value for different usecases (E.g. MSGTYPE=1 for Plain text in Push Notifications, Refer below table for more details)
<b>ADDRESS</b> Describe the Sender as well as Receiver address	
FROM	Business WhatsApp Number
TO	Receivers WhatsApp Number
SEQ	Unique Sequence ID. Must be an integer and must be unique to each WhatsApp message. While checking message status, this value can be used.
TAG	This is an optional parameter, which can be used to send any additional information, which then stored at ValueFirst's end, and can be retrieved at a later date or can be displayed in MIS.
TEMPLATEI NFO	Template ID and variable separated by ~ (E.g.: TEMPLATEINFO="Templateid~v1~v2~v3")
MSG-TYPE	MSG-TYPE=3 indicates that the message contains Rich Media in Push Notification
MEDIADATA	URL where the media or file is located.

TYPE	Media Type E.g.: TYPE=" image" (Possible values are image, document) For document, user can send the name also. Eg:- <b>TYPE="document~filename to be displayed"</b>
B_URLINFO	Variable value for URL button of Call to Action

CH_TYPE	MSGTYPE	Action Description
4	1	Plain text/template id in Push Notification
4	2	Plain text in Two Way
4	3	Rich media in Push notification using media URL
4	4	Rich media in Two Way using media URL
4	5	Business Card sharing in Two Way
4	6	Location sharing in Two Way

### Example of WhatsApp Submission Response (XML API)

```
<MESSAGEACK>
  <GUID GUID="kb4jc4243489a5f461209eo742--TESTUSER" SUBMITDATE="2017-05-10 09:59:49" ID="1">
    <ERROR SEQ="2" CODE="28682"/>
  </GUID>
  <GUID GUID="kb4jc4334130a5f461209l7t17--TESTUSER" SUBMITDATE="2017-05-10 09:59:49" ID="2"/>
  </GUID>
</MESSAGEACK>
```

### Example of WhatsApp Submission Response (JSON API)

```
{
  "MESSAGEACK": {
    "GUID": {
      "SUBMITDATE": "2020-01-10 13:34:58",
      "GUID": "kk1ad345819911f440000ree6nDEMOSOUMYA",
      "ID": 1
    }
  }
}
```

The following table describes different elements of WhatsApp Response.

Tag Name	Description
GUID	A globally unique message ID that is generated for each <SMS> tag. This GUID must be stored at user's application, for them to be able to check the
SUBMITDATE	The date and time when the transaction was completed.
ID	Unique SMS ID sent by the user. For each message a unique GUID is generated. Server sends SMS ID so that the client's application can map the

ERROR	<p><b>(In case of any error)</b></p> <p>To conserve bandwidth utilization ValueFirst WhatsApp API only sends sequence information of messages that has either some error or was rejected because of some error.</p> <p>If there are no errors in a particular message, you shall not receive any confirmation of each address SEQ. For instance, in the above example in message ID 1 (of client) the TO number „My company“ was rejected as non-numeric. The second message does not have any error, and hence there was no error information for the second part.</p>
-------	---

Please refer to error code description table at the end of this document for more details on available error codes.

### WhatsApp Status Request Service

Status request API supports multiple message status per transaction. A simple example of WhatsApp Status request is shown below:

#### XML API

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE STATUSREQUEST SYSTEM
"http://127.0.0.1:80/psms/dtd/requeststatusv12.dtd">
<STATUSREQUEST>
  <USER USERNAME="mycompany" PASSWORD="mycompany"/>
  <GUID GUID="kb4jc5342283a5f461209c6m6h--TESTUSER">
    <STATUS SEQ="1" />
    <STATUS SEQ="2" />
  </GUID>
  <GUID GUID="kb4jc5407304a5f461209c1x89--TESTUSER" />
</STATUSREQUEST>
```

#### JSON API

```
{
  "VER": "1.2",
  "STATUSREQUEST": {
    "USER": {
      "@USERNAME": "mycompany",
      "@PASSWORD": "mycompany",
      "@UNIXTIMESTAMP": ""
    },
    "GUID": [
      {
        "@GUID": "kk1ad345819911f440000ree6nDEMOSOUMYA",
        "STATUS": [
          {
            "@SEQ": ""
          },
          {
            "@SEQ": ""
          }
        ]
      }
    ]
  }
}
```

```
} ] } ]}]}
```

The elements of the above example are explained in the following table.

Tag Name	Description
GUID	A globally unique message ID that is generated for each <SMS> tag. This GUID is generated when ValueFirst Pace receives a new session.
Seq	SEQ ID of Mobile numbers whose status needs to be extracted. If no Status tag is sent the API shall return status of all Sequences in the specified Transaction/GUID.
UserName	API username
Password	API password

When the server receives WhatsApp-SR query, following responses would be returned.

### XML API

```
<STATUSACK>
  <GUID GUID="kb4jc5441583a5f461209hmvdf--TESTUSER">
    <STATUS SEQ="1" ERR="8448" DONEDATE="2017-05-10 10:06:31"
    REASONCODE="000">
      <STATUS SEQ="2" ERR="8448" DONEDATE="2017-05-10 10:06:31"
      REASONCODE="000" >
    </GUID>
    <GUID GUID="kb4jc5540144a5f461209het83--TESTUSER">
      <STATUS SEQ="1" ERR="8448" DONEDATE="2017-05-10
10:06:31" REASONCODE="000"/>
      <STATUS SEQ="2" ERR="8449" DONEDATE="2017-05-10
10:06:37" REASONCODE="009"/>
    </GUID>
  </STATUSACK>
```

### JSON API

```
{
  "STATUSACK": {
    "GUID": {
      "STATUS": {
        "DONEDATE": "2020-01-10 13:34:59",
        "REASONCODE": 173,
        "ERR": 8448,
        "SEQ": 1
      },
      "GUID": "kk1ad345819911f440000ree6nDEMOSOUMYA"
    }
  }
}
```

The elements of the above response are explained in the following table.

Tag Name	Description
GUID	A globally unique Message ID that is generated for each <SMS> tag.

SEQ	SEQ ID of Mobile numbers whose status needs to be extracted.
DONEDATE	The time when latest status was received. The new status could be either success or failure, the field is in Standard ANSI format, i.e., YYYY-Error / Message Status Code, if no standard error occurred, the ERR shall be either one of the following values. 8448: Message was successfully delivered on DONEDATE 8449: Message reportedly failed on DONEDATE
ERR	Error code for failure or success
REASONCODE	Reason-code is an optional variable.

Please refer to error code description table at the end of this document for more details on available error codes.

### WhatsApp Credit Request Service

Credit request API can be used to check credit status for the API user. A simple example of credit request API is shown below:

#### Example of WhatsApp Credit Request (FOR XML API)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE REQUESTCREDIT
"http://127.0.0.1:80/psms/dtd/requestcredit.dtd">
<REQUESTCREDIT USERNAME="" PASSWORD="">
</REQUESTCREDIT>
```

**SYSTEM**

#### Example of WhatsApp Credit Request (FOR JSON API)

```
{
  "REQUESTCREDIT": {
    "@USERNAME": "mycompany",
    "@PASSWORD": "mycompany"
  }
}
```

The elements of the above example are explained in the following table:

Tag Name	Description
UserName	API username
Password	API password

#### Example of WhatsApp Credit Request Response (XML API)

```
<SMS-Credit User="test">
<Credit Limit="100000000" Used="1522932.00"/>
</SMS-Credit>
```

```
{
  "SMS-Credit": {
    "User": "mycompany",
    "Credit": {
      "Used": "618.00",
      "Limit": "1028"
    }
  }
}
```



}

The elements of the above XML are explained in the following table:

<b>Credit Limit</b>	Total credits
<b>Used</b>	Credits used.

## DLR URL

DLR URL functionality is provided for users to be able to receive delivery reports in real-time against each message submitted using ValueFirst APIs. Users can pass DLR URL along with the SMS packet on which ValueFirst API will send the DLR as soon as it is received at ValueFirst's end. Below is the sample XML format for passing DLR URL

Below is a sample XML with DLR URL tag:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE MESSAGE SYSTEM "http://127.0.0.1:80/psms/dtd/messagev12.dtd" >
<MESSAGE>
<USER USERNAME="mycompany" PASSWORD="mycompany" CH_TYPE="4" />
< DLR URL="http://myurl.domain/recvdlr?dlrparameters" <SMS UDH="0" CODING="1"
TEXT="hey this is a real test" PROPERTY="" MSGTYPE="1" ID="11">
<ADDRESS FROM="91XXXXXXXXXX" TO="91XXXXXXXXXX" SEQ="1" />
</SMS>
</MESSAGE>
```

DLR URL for WhatsApp can be configured at the backend, by our support team against a username on which ValueFirst API will send the DLR as soon as the status is received.

### Sample URL for DLR:

Parameters are optional can be configured as per the requirement.

GET/Post Both Supported and data is passed in query params manner.

Example :

**[https://sampleurl.com/WAstatus?TO=91XXXXXXXXXX&FROM=91XXXXXXXXXX&MESSAGE\\_STATUS=2&REASON\\_CODE=401&CLIENT\\_GUID=kk2j95XXXXX&MESSAGE\\_ID=k2XXXXX&STATUS\\_ERROR=8449](https://sampleurl.com/WAstatus?TO=91XXXXXXXXXX&FROM=91XXXXXXXXXX&MESSAGE_STATUS=2&REASON_CODE=401&CLIENT_GUID=kk2j95XXXXX&MESSAGE_ID=k2XXXXX&STATUS_ERROR=8449)**

Below table depicts the parameter values to be passed in DLR URL to get corresponding values.

Sample Parameter names	Values	Sample Response
TO	%p	919812345678
FROM	%P	Business WhatsApp Number
TIME	%t	2017-05-23 16:55:18
MESSAGE_STATUS	%d	1
REASON_CODE	%2	000
DELIVERED_DATE	%3	2017-05-23 16:55:18

STATUS_ERROR	%4	8448
CLIENT_GUID	%5	kh5ng551155213b161011bme3vTESTUSER
CLIENT_SEQ_NUMBER	%6	Value as Submitted
MESSAGE_ID	%7	h5ng551155313946013uw3
CIRCLE	%8	Delhi
OPERATOR	%9	AIRCEL
TEXT_STATUS	%13	Success
SUBMIT_DATE	%14	2017-05-23 16:55:11
MSG_STATUS	%16	Delivered
TAG	%TAG	Some client side data
Additional Parameter 1	%21	Value as Submitted
Additional Parameter 2	%22	Value as Submitted
Additional Parameter 3	%23	Value as Submitted
Additional Parameter 4	%24	Value as Submitted
Additional Parameter 5	%25	Value as Submitted
Button Label	\$btnLabel	Label as configured in Template
Button Clicked Date time	\$clickedOn	Button Click Date Time Ex -2017-05-23 16:55:18

**Note:**

In the requested DLR URL, the attributes used in the URL can be named accordingly but values used corresponding to used attributes or variables are case sensitive.

**WhatsApp DLR Pull**

Possible delivery status for WhatsApp messages can be Invalid contact, Sent, Delivered or Read. Below are the values for MESSAGE\_STATUS

- 0: When message has been submitted to WhatsApp
- 1: In case of read or delivered status is received from WhatsApp.

Note: Read and delivered can be differentiated basis REASONCODE and ERR value combination. Example for the same is mentioned below:

DLR for Delivered but not read

**XML**

```
<STATUSACK>
  <GUID GUID="kj7nj3213005h1f410000t7qpxDEMOSOUMYA">
    <STATUS SEQ="1" ERR="8448" DONEDATE="2019-07-23 19:32:13"
    REASONCODE="173" CHANNEL="4"/>
  </GUID>
</STATUSACK>
```

**JSON**

```
{
  "STATUSACK": {
    "GUID": {
      "STATUS": {
        "DONEDATE": "2020-01-10 17:32:54",
        "REASONCODE": "173",
        "ERR": 8448,
```

```

    "SEQ": 1
  },
  "GUID": "kk1ah3241220f1f4400005u2vqTRUEBILPAI"
}
}
}
}
DLR for Read
XML
<STATUSACK>
  <GUID GUID="kj7nj3213005h1f410000t7qoxDEMOSOUMYA">
    <STATUS SEQ="1" ERR="8448" DONEDATE="2019-07-23 19:32:40"
    REASONCODE="000" CHANNEL="4"/>
  </GUID>
</STATUSACK>

```

### JSON

```

{
  "STATUSACK": {
    "GUID": {
      "STATUS": {
        "DONEDATE": "2020-01-10 17:32:54",
        "REASONCODE": "000",
        "ERR": 8448,
        "SEQ": 1
      },
      "GUID": "kk1ah3241220f1f4400005u2vqTRUEBILPAI"
    }
  }
}

```

### Standard API Error Codes

Below is the list of available error codes in ValueFirst WhatsApp API.

Error Code	Description
<b>General</b>	
52992	Username / Password incorrect
57089	Contract expired
57090	User credit expired
57091	User disabled
65280	Service is temporarily unavailable
65535	The specified message does not conform to DTD
0	WhatsApp submitted success NO Error (Not returned in ValueFirst Whatsapp API )
<b>Message Post</b>	
28673	Destination number not numeric
28674	Destination number empty
28675	Sender address empty
28676	Template mismatch
28677	UDH is invalid / SPAM message
28678	Coding is invalid

28679	SMS text is empty
28680	Invalid sender ID
28681	Invalid message, Duplicate message, Submit failed
28682	Invalid Receiver ID (Will validate Indian mobile numbers only.)
28683	Invalid Date time for message Schedule (If the date specified in message post for schedule delivery is less than current date or more than expiry date or more than 1 year)
28692	Invalid Split Count
28694	Invalid/Incomplete details in TEMPLATEINFO tag of request
28695	Template matched but invalid variables in template.
28696	Media ID, type, content type cannot be null or blank in case of two way media message. If text have data, it should be in base64 format.
28697	Incorrect User Route
28698	Invalid/Blank Media data or type(In case of Push Media Message)
28699	MSGTYPE in WhatsApp Data Request Packet is empty.
28700	WhatsApp Data Request Packet is Empty / Media type in WhatsApp data request packet does not match with the MSGTYPE received in the request.
28702	Invalid DLT Parameters
28703	Invalid DLT Content Type
28704	Invalid Authorization Type(If message is rejected due to authorization scheme other than Authorization header)
	<p><b>Note:</b> If message is rejected for wrong Authorization method selected then that message is rejected on filter with Http error code 401 and description <b>"Unauthorized"</b></p> <p>If message is rejected for HTTPS then that message is rejected on filter with Http error code 403 and description <b>"HTTP Not Allowed"</b></p>
<b>Status Request</b>	
8448	Message delivered successfully
8449	Message failed
<b>Reason Codes</b>	
000	Read
173	Delivered but not Read
401	Contact not registered on WhatsApp
402	Sent
470	Message failed to send because more than 24 hours have passed since the customer last replied to this number
1014	Upload failed on bad image (image not uploaded correctly) or endpoint not found
2008	Media format used is unsupported
2009	Required component in the Template is missing
2010	URL in button component is invalid
2011	Phone Number in button component is invalid
2012	Parameter format does not match format in the created Template

2013	Buttons are unsupported by the receiver
100	Miscellaneous
Others	<a href="https://developers.facebook.com/docs/whatsapp/api/errors">https://developers.facebook.com/docs/whatsapp/api/errors</a>

**Note:** No status will be returned for messages, which are in waiting status.

## WhatsApp Two Way Forwarding API / Incoming API

### HTTP Methods available

1. GET : Supports query parameters (mentioned below) in the URL and any 5 headers.
2. POST : Supports parameters (mentioned below) in both query and body in raw or JSON format and any 5 headers.

### Suggested URL

[http://IP:Port/?WABANumber=\\$WABANumber&FromNumber=\\$FromNumber&ReceiveDate=\\$ReceiveDate&ExpiryDate=\\$ExpiryDate&Text=\\$Text&Caption=\\$Caption&MediaType=\\$MediaType&EndUserName=\\$EndUserName&Latitude=\\$Latitude&Longitude=\\$Longitude&ContentType=\\$ContentType&MediaData=\\$MediaData&Userid=\\$Userid&FrequentlyForwarded=\\$frequentlyForwarded&Forwarded=\\$forwarded&Profile=\\$Profile&City=\\$City&Country=\\$Country&CountryCode=\\$CountryCode&State=\\$State&Street=\\$Street&AddressType=\\$AddressType&Zip=\\$Zip&Email=\\$Email&FirstName=\\$FirstName&Middlename=\\$Middlename&LastName=\\$LastName&ButtonLabel=\\$btnLabel&FormattedName=\\$FormattedName&Phone=\\$Phone&PhoneType=\\$PhoneType&Company=\\$Company&OrgTitle=\\$OrgTitle](http://IP:Port/?WABANumber=$WABANumber&FromNumber=$FromNumber&ReceiveDate=$ReceiveDate&ExpiryDate=$ExpiryDate&Text=$Text&Caption=$Caption&MediaType=$MediaType&EndUserName=$EndUserName&Latitude=$Latitude&Longitude=$Longitude&ContentType=$ContentType&MediaData=$MediaData&Userid=$Userid&FrequentlyForwarded=$frequentlyForwarded&Forwarded=$forwarded&Profile=$Profile&City=$City&Country=$Country&CountryCode=$CountryCode&State=$State&Street=$Street&AddressType=$AddressType&Zip=$Zip&Email=$Email&FirstName=$FirstName&Middlename=$Middlename&LastName=$LastName&ButtonLabel=$btnLabel&FormattedName=$FormattedName&Phone=$Phone&PhoneType=$PhoneType&Company=$Company&OrgTitle=$OrgTitle)

**Note :** We will forward all the values received from WhatsApp as per the messages type and rest will remain as it is. For ex. We will replace only to, from, text, received date in case of text rest parameters like \$ContentType, \$Latitude etc will be sent as it is.

The parameters available are as following:

WABANumber	\$WABANumber	918744960111 (Brands WABA Number)
FromNumber	\$FromNumber	91XXXXXXX (End users Number)
ReceiveDate	\$ReceiveDate	2020-01-10 17:32:54 (Received Data)
ExpiryDate	\$ExpiryDate	2020-01-11 17:32:54
Text	\$Text	Test Message
Caption	\$Caption	Test Caption
MediaType	\$MediaType	document
EndUserName	\$EndUserName	John Doe
Latitude	\$Latitude	28.4309227
Longitude	\$Longitude	77.0136975
ContentType	\$ContentType	pdf
MediaData	\$MediaData	Base64 Encoded data

Userid (Client userid in VF)	\$Userid	Demoabc (Client userid at ValueFirst)
FrequentlyForwarded	\$frequentlyForwarded	True (If end user sends a forwarded message)
Forwarded	\$forwarded	True (If end user sends a frequently forwarded message)
Profile	\$Profile	Abc
City	\$City	Gurgaon
Country	\$Country	India
CountryCode	\$CountryCode	+91
State	\$State	Haryana
Street	\$Street	Wall Street
AddressType	\$AddressType	Work
Zip	\$Zip	122001
Email	\$Email	john@doe.com
FirstName	\$FirstName	ValueFirst Reception
Middlename	\$Middlename	VF
LastName	\$LastName	VF
Button Label	\$btnLabel	Label name as configured in WhatsApp Template
FormattedName	\$FormattedName	VF Admin
Phone	\$Phone	91XXXXXXXXXX (Shared contact number)
PhoneType	\$PhoneType	Mobile
Company	\$Company	ValueFirst
OrgTitle	\$OrgTitle	Mr

Media	Supported Content-Types
<code>document</code>	<code>application/pdf,</code> <code>application/msword,</code> <code>application/vnd.ms-powerpoint,</code> <code>application/vnd.ms-excel,</code> <code>text/plain</code>
<code>image</code>	<code>image/jpeg,</code> <code>image/png</code>
<code>audio</code>	<code>audio/acc,</code> <code>audio/mp4,</code> <code>audio/amr,</code> <code>audio/mpeg,</code> <code>audio/ogg,</code> <code>codecs=opus</code>
<code>video</code>	<code>video/mp4</code> <b>Note:</b> Only H.264 video codec and AAC audio codec is supported.

## Regulatory Implementation and Impact

WhatsApp does not allow use of the platform for promotional messages, product information, and loyalty or cross selling as of now.

1. Customer will upload two separate files to the FTP location. Extension of these two file should be "\*.done" and "\*.txt".
2. TXT file should contain the data and .done file should be of "0KB" which will be searched for executing any campaign.
3. ValueFirst will process the .txt file if its corresponding .done file is available on the FTP path.

Ex for illustration:

We have received below Files on our FTP Servers

ABC.txt	ABC.done
XYZ.txt	XYZ.done
SMS.txt	<b>"-NO .doneFile"</b>

In this case VF will process only "ABC.txt" and "XYZ.txt" as for "SMS.TXT" we have not received "SMS.done" file.

4. TXT files format should be "|" delimited without header. Thus the data that it should Mobile Number & Text.

919XXXXXXXXXX|You have made a purchase XXX XXX XXX

919XXXXXXXXXX|You have made a selling for XXX XXX XXX

919XXXXXXXXXX|You have made purchase of amt XXX XXX XXX

After processing the file will be moved in another folder on same location. Name of the folder will be ARCHIVE.

e.g. a file is received in path\FolderX; it will be moved to path\FolderX\ARCHIVE.

5. If in case at the time of polling the files from the FTP location, any txt file is found to be a lone file without its consecutive .done an alert will be generated to user.
6. Report will be sent to FTP on daily basis on the same location from which user will process the file.

File format-CSV.

A new folder will be created for dlrlog at same FTP location \FolderX\DLRLOGS

GUID	Recipient	Sender	Sent_On	Received_On	Status	SMS_Split	Reason_of_failure	Operator
------	-----------	--------	---------	-------------	--------	-----------	-------------------	----------

File will be uploaded next day containing the data of previous day (once in a day).