Ved Tech Labs API User Manual

TABLE OF CONTENT

1. Introduction

The Ved Tech Labs API is a comprehensive SMS messaging platform that provides developers with multiple interfaces to send SMS messages. The API supports HTTP/HTTPS protocols and offers three main integration methods: PUSH API, XML Interface, and JSON API.

This manual covers all aspects of the Ved Tech Labs API, including authentication, parameter specifications, request/response formats, and error handling. The API is designed to handle both single and bulk SMS messaging with support for plain text and Unicode messages.

Key Features:

- Multiple API interfaces (HTTP, XML, JSON)

- Support for both GET and POST methods
- Plain text and Unicode message support
- Bulk messaging capabilities
- Short URL generation and tracking
- DLT compliance with Entity ID and Template ID support

2. Authentication

The Ved Tech Labs API uses a combination of username and API key for authentication. All requests must include valid credentials to access the messaging services.

Required Authentication Parameters:

- **username:** Your account identifier provided by the SMS platform
- **apikey:** Your unique API key for request authentication
- **signature:** Your registered Sender ID

Note: Your username and API key will be provided by the SMS platform administrator upon account setup. Keep your API key secure and do not share it publicly.

3. Base URL & Endpoints

The Ved Tech Labs API supports both HTTP and HTTPS protocols. For security purposes, HTTPS is recommended for all production implementations.

> **Base URL:** https://your-domain.com/pushapi/

Available Endpoints:

Endpoint	Method	Description
/sendmsg	GET	Send single SMS message
/sendbulkmsg	GET	Send bulk SMS messages
/sendxmlmsg	POST	Send SMS via XML interface
/json/sendbulkmsg	POST	Send bulk SMS via JSON interface

4. PUSH API

The PUSH API is the primary interface for sending SMS messages. It supports both GET and POST methods and provides comprehensive parameter options for message customization.

Parameter List:

1	username	String	Required	All	Account id; will be provided by SMS Platform
2	dest	String	Required	All	The target mobile number on which the SMS is being sent
3	apikey	String	Required	All	User specific key for every request
4	signature	String	Required	All	Sender ID
5	msgType	String	Required	All	Type of message (PM for plain text or UC for Unicode message/multilingual message)
6	msgText	String	Required	All	Text to be sent

S.No	Parameter	Type	Required/Optional	Use	Description
7	custref	String	Optional	All	Customer reference key specific to campaign
8	entityid	Numeric	Optional	All	DLT register Entity ID for the Header
9	templateid	Numeric	Optional	All	DLT register Template ID for the Message Template
11	domain	String	Optional	Short URL	User domain to be used in short URL but that should be approved in platform
12	converturl	String	Optional	Short URL	This flag can be used. if user don't want to convert URL then use this with N. By default its value is Y.
13	campaign	String	Optional	Short URL	Campaign name for the campaign, if not provided. Then it will be autogenerated by system for the day and can be used for tracking.
14	crmurl	String	Optional	Short URL	callback url which can be used for short url tracking. Passing the clicking values

Sample URL Format:

```
Method Type - GET

Endpoint : https://xyz.com/pushapi/sendmsg?
username=xxxxxx&dest=xxxxxxx&apikey=12345&signature=SENDERID&msgtype=PM&msgtx
t=message&entityid=XXXXXXXXXXX&templateid=XXXXXXXXXXXXX
```

Message Type Examples:

Plain Text Message:

```
http://domain/pushapi/sendmsg?
username=xxxxxxx&dest=xxxxxxxxxx&apikey=12345&signature=SENDERID&msgtype=PM&m
sgtxt=message&custref=xxxxx
```

Unicode/Multi-Lingual Message:

```
http://domain/pushapi/sendmsg?
username=xxxxxx&dest=xxxxxxxxxx&apikey=12345&signature=SENDERID&msgtype=UC&ms
gtxt=message&custref=xxxxx
```

Bulk SMS:

```
Method Type : GET

Sample Request : https://domain.com/pushapi/sendbulkmsg?
username=xxxxxx&dest=xxxxxxxxxx,xxxxxxxxxx,xxxxxxxxxx,xxxxxxxxxx&apikey=12345
&signature=SENDERID&msgtype=PM&msgtxt=message&entity id=XXXXXXXXXX&templatei
```

Note: For bulk SMS, you can add multiple mobile numbers separated by commas in the dest parameter.

5. XML Interface

The XML Interface provides a structured way to send SMS messages using XML format. This interface supports POST method only and is ideal for complex messaging scenarios.

Supported Method:

POST

Endpoint:

```
Method : POST
Endpoint : https://yourdomain.com/pushapi/sendxmlmsg
Content-Type : application/xml
```

XML Structure:

The XML interface supports three messaging patterns:

- **One to One:** Send a single message to a single recipient

```
<xmlPushRequest username="XXXX" pin="XXXX">
   <campaignName>XXXX</campaignName>
   <entityId>XXXXXXXXXXXX</entityId>
   <data>
      <from>XXXX</from>
      <recipientList>
         <recipient>
            <destAddress>XXXXXXXXXX</destAddress>
            <custref>XXXX</custref>
         </recipient>
      </recipientList>
      <message>
         <templateId>XXXXXXXXXXXX</templateId>
         <messageType>PM</messageType>
         <messageTxt>Your message text goes here.</messageTxt>
      </message>
   </data>
</xmlPushRequest>
```

- **One to Many:** Send the same message to multiple recipients

```
<xmlPushRequest username="XXXX" pin="XXXX">
    <campaignName>XXXX</campaignName>
    <entityId>XXXXXXXXXXXX</entityId>
    <data>
        <from>XXXX</from>
        <recipientList>
            <recipient>
                <destAddress>XXXXXXXXXX</destAddress>
                <custref>XXXX</custref>
            </recipient>
            <recipient>
                <destAddress>XXXXXXXXXX</destAddress>
                <custref>XXXX</custref>
            </recipient>
            <recipient>
                <destAddress>XXXXXXXXXX</destAddress>
                <custref>XXXX</custref>
            </recipient>
        </recipientList>
        <message>
            <templateId>XXXXXXXXXXXX</templateId>
            <messageType>PM</messageType>
            <messageTxt>Your message text goes here.</messageTxt>
        </message>
    </data>
</xmlPushRequest>
```

- **Many to Many:** Send different messages to different recipients

Content-Type: application/xml

Method: POST

Encoding: UTF-8

```
<xmlPushRequest username="XXXX" pin="XXXX">
    <campaignName>XXXX</campaignName>
    <entityId>XXXXXXXXXXXX</entityId>

    <data>
        <from>XXXX</from>
        <recipientList>
            <recipient>
                <destAddress>XXXXXXXXXX</destAddress>
                <custref>XXXX</custref>
            </recipient>
        </recipientList>
        <message>
            <templateId>XXXXXXXXXXXX</templateId>
            <messageType>PM</messageType>
            <messageTxt>Message text A</messageTxt>
        </message>
    </data>

    <data>
        <from>XXXX</from>
        <recipientList>
            <recipient>
                <destAddress>XXXXXXXXXX</destAddress>
                <custref>XXXX</custref>
            </recipient>
        </recipientList>
        <message>
            <templateId>XXXXXXXXXXXX</templateId>
            <messageType>PM</messageType>
            <messageTxt>Message text B</messageTxt>
        </message>
    </data>
```

```
</xmlPushRequest>
```

6. JSON API

The JSON API provides a modern RESTful interface for sending bulk SMS messages. It uses JSON format for both request and response, making it easy to integrate with modern applications.

- **One to One**

```
Endpoint : https://domainapi/pushapi/json/sendbulkmsg
Method Type : POST
Content-Type : application/json
```

```
{
  "username": "XXXX",
  "password": "XXXX",
  "senderid": "XXXX",
  "entityid": "XXXXXXXXXXXX",
  "campaignname": "XXXX",
  "smslist": [
    {
      "text": "Your OTP is 123456. Do not share it.",
      "mobiles": "91XXXXXXXXXX"
      "messagetype": "PM",
      "custref": "XXXX1",
      "templateid": "XXXXXXXXXXXX"
    }
  ]
}
```

- **One to Many**

```
Endpoint : https://domainapi/pushapi/json/sendbulkmsg
Method Type : POST
Content-Type : application/json
```

```
{
  "username": "XXXX",
  "password": "XXXX",
  "senderid": "XXXX",
  "entityid": "XXXXXXXXXXXX",
  "campaignname": "XXXX",
  "smslist": [
    {
      "text": "Your OTP is 123456. Do not share it.",
      "mobiles": "91XXXXXXXXXX,91XXXXXXXXXX",
      "messagetype": "PM",
      "custref": "XXXX1",
      "templateid": "XXXXXXXXXXXX"
    }
  ]
}
```

- **Many to Many**

```
Endpoint : https://domainapi/pushapi/json/sendbulkmsg
Method Type : POST
Content-Type : application/json
```

```
{
  "username": "XXXX",
  "password": "XXXX",
  "senderid": "XXXX",
  "entityid": "XXXXXXXXXXXX",
  "campaignname": "XXXX",
  "smslist": [
    {
      "text": "Your Message Text 1 Here.",
      "mobiles": "91XXXXXXXXXX,91XXXXXXXXXX",
      "messagetype": "PM",
      "custref": "XXXX1",
      "templateid": "XXXXXXXXXXXX"
    },
    {
      "text": " Your Message Text 2 Here.",
      "mobiles": "91XXXXXXXXXX,91XXXXXXXXXX",
      "messagetype": "PM",
      "custref": "XXXX2",
      "templateid": "XXXXXXXXXXXX"
    }
  ]
}
```

7. Response Formats

The Ved Tech Labs API returns structured JSON responses for all requests. Response formats vary based on the type of message and API interface used.

Success Response Structure:

Sample Success Response:


```
{ "code": "6001", "desc": "Message received by platform.", "reqId":
"033225001159310350274419 9", "time": "2023-02-25 22:15:02.744", "custRef":
"12345", "partMessageIds": ["033225001159310350274419 9"],
"totalMessageParts": 1, "campaignName": "test" }
```


Success Response with Short URL:


```
{ "code": "6001", "desc": "Message received by platform.", "reqId":
"033225001159307667219133 0", "time": "2023-02-25 14:47:52.191", "custRef":
"12345", "longUrl": "http://www.google.co.in", "shortUrl": "gmly.in/cL",
"partMessageIds": ["033225001159307667219133 0"], "totalMessageParts": 1,
"campaignName": "test" }
```


Response Fields Explanation:

Field	Type	Description
code	String	Response code indicating success or failure
desc	String	Human-readable description of the response
reqId	String	Unique request identifier for tracking
time	String	Timestamp of the request processing
custRef	String	Customer reference provided in the request
partMessageIds	Array	Array of message part IDs for multipart messages
totalMessageParts	Number	Total number of message parts
campaignName	String	Name of the campaign
longUrl	String	Original long URL (if URL shortening is used)
shortUrl	String	Generated short URL

CRM URL Callback Format:

```
https://yourdomain.com/callback?reqid=##messageid##&mob=##mobile##&reqtime=##submittime##&deliverytime=##deliverytime##&deliverycode=##status##&status=##deliverycode##
```

8. Error Codes

The Ved Tech Labs API uses standardized error codes to indicate different types of failures. Understanding these codes helps in debugging and handling errors appropriately.

Sample Error Response:

```
{ "code": "500", "desc": "Internal error occurred.", "reqId": null, "time": null, "custRef": null }
```

Common Error Codes:

Error Code	Description	Resolution
500	Internal error occurred	Contact support team for assistance
114	Invalid API Key	Invalid API key
110	Invalid destination number	Verify mobile number format (include country code)
117	Invalid message text	Provide valid message content
105	Invalid sender ID	Use approved sender ID from your account
6001	Message received by platform (Success)	Message successfully queued for delivery

Note: Positive codes (like 6001) indicate success, while negative codes indicate various types of errors.

9. Contact Support

Support Type	Contact Method	Response Time	Availability

Our technical support team is available to assist you with integration, troubleshooting, and optimization of your SMS messaging implementation.

Before Contacting Support:

- Check your API credentials and permissions
- Review error codes and messages
- Test with sample requests provided in this manual
- Gather relevant logs and error messages
- Prepare your account details and use case information

Business Hours: Monday to Friday, 9:00 AM to 6:00 PM (IST)
Emergency Support: Available 24/7 for critical production issues

---