VASBAY

SMS API & SMPP Reference Guide

Send SMS, check credits and fetch delivery reports over HTTP (REST) or SOAP. URL parameters, JSON, XML or SOAP in; JSON, XML or SOAP out.

Version	3.0
Last updated	4 Nov 2026
Base URL	https://www.vasbay.com/webservice/
SOAP / WSDL	https://www.vasbay.com/webservice/soap.php?wsdl
Online version	https://www.vasbay.com/api-documentation

This document is intended for developers integrating with the VASbay SMS platform. Keep your API key confidential.

Contents

1	Introduction	Overview and endpoints
2	Authentication	Credentials, domain and IP rules
3	Request and response formats	Input styles, format=, envelopes
4	API v3 (JSON)	/api/v3, sign-in, endpoints, HTTP status
5	Mobile apps	App sign-in with tokens
6	SDKs and plugins	JavaScript, PHP, Python, Laravel, WordPress, Magento
7	Send SMS (classic API)	smsSimpleApi.php
8	Bulk and dynamic SMS	smsApi.php
9	Scheduled SMS	scheduleType / scheduleTime
10	Credit balance	smsBalanceApi.php
11	Delivery status	smsStatusApi.php
12	SOAP API	soap.php, WSDL, operations
13	Error codes	Every code and message
14	Code samples	PHP, Python, Node.js, Java
15	Notes and limits	Templates, credits, security
A	Appendix: quick reference	All calls on one page
B	Appendix: SMPP connectivity	Binds, submit_sm, DLT TLVs, receipts, status codes

1 Introduction

The VASbay SMS API lets your website, mobile back-end, CRM, ERP or billing software send transactional and promotional SMS, check the credits left on your account and fetch delivery reports.

The same API answers **REST** calls (URL parameters, form posts, JSON or XML bodies) and **SOAP 1.1** calls, and replies in **JSON** (default), **XML** or **SOAP**.

Base URL	`https://www.vasbay.com/webservice/`
Input	URL / form parameters, JSON, XML, SOAP 1.1 envelope
Output	`format=json` (default), `xml` or `soap`
Encoding	UTF-8 for requests and responses
Transport	HTTPS recommended

Endpoints

ENDPOINT	METHOD	PURPOSE
`smsSimpleApi.php`	GET · POST	Send one message to one or more numbers
`smsApi.php`	POST	Bulk send with your own reference per number, and dynamic (personalised) messages
`smsBalanceApi.php`	GET · POST	Credits left per route
`smsStatusApi.php`	GET · POST	Delivery status of a push or a single number
`soap.php`	POST (WSDL: GET `?wsdl`)	SOAP operations `sendSms`, `getBalance`, `getStatus`

Older integrations keep working unchanged: `sentsms.php` behaves as `smsSimpleApi.php` / `smsApi.php`, `smsBalance.php` as `smsBalanceApi.php` and `smsStatus.php` as `smsStatusApi.php`.

2 Authentication

Every call carries two credentials, as parameters or inside the body (JSON `head`, XML `<head>` or the SOAP operation element).

FIELD	DESCRIPTION
`usersName`	Your UID: your login name, hex-encoded (for example the login `admin` is `61646d696e`). Shown in the panel under **My Profile › API Access**.
`key`	An enabled API key from **My Profile › API Access**. Keep it secret; disable it there at once if it leaks and create a new one.

Rules

- **Use your own panel's domain.** Accounts belong to the panel (domain) they were created on, and the API only accepts them on that domain: `www.vasbay.com` for this panel. A call on another domain is refused with `601`.
- **Allowed IP addresses.** When allowed IPs are set on a key, calls from any other address are refused with `601`. Several addresses are separated by commas.
- **One answer for every failure.** A wrong or disabled key, an inactive account, a wrong domain or an address that is not allowed all return `601 Authentication Failed`, so the response never reveals which part was wrong.

3 Request and response formats

3.1 Sending a request

STYLE	HOW
URL parameters	`GET …/smsSimpleApi.php?usersName=…&key=…&…` – values URL-encoded
Form post	`POST` with `Content-Type: application/x-www-form-urlencoded` (or multipart)
JSON	`POST` a JSON object: flat (`{"route":"2","numbers":["98…","97…"],…}`) or `{"smsRequest":{"head":{…},"request":[…]}}`
XML	`POST` `<smsRequest><head>…</head><request>…</request></smsRequest>`

STYLE	HOW
SOAP 1.1	`POST` an envelope to `soap.php`

URL parameters and body values may be mixed, for example credentials in the URL and the message in the body. A value in the body overrides the same parameter in the URL. XML bodies may not contain a `DOCTYPE` or entity declarations.

3.2 Choosing the response format

Add `format` to the request (URL parameter or body field). Without it, REST calls answer in JSON and SOAP calls in SOAP. Any other value returns error `603`.

FORMAT	CONTENT-TYPE	SHAPE
`json` (default)	`application/json`	the objects shown in this document
`xml`	`application/xml`	same fields under `<result>`; numeric keys become `<n0>`, `<n1>` …
`soap`	`text/xml`	the XML result inside `<soap:Envelope><soap:Body><{operation}Response>`

3.3 Success and error envelopes

- **Send SMS** and **delivery status** succeed with `"data": "0"`; the result is in `msg`.
- **Credit balance** succeeds with `"status": "0"`; the result is in `response`.
- **Every error** has a non-zero `status` code and the message in `response`.
- The HTTP status is `200` in all cases; always check the code in the body.

ERROR · JSON

```
{ "status": "601", "response": "Authentication Failed" }
```

ERROR · XML

```
<?xml version="1.0"?>
<result><status>601</status><response>Authentication Failed</response></result>
```

4 API v3 (JSON)

A JSON REST API with one set of endpoints for servers, the SDKs and plugins, and mobile apps. Replies are always JSON with proper HTTP status codes. Use it for new integrations; the classic API (chapters 7 to 12) keeps working.

BASE `https://www.vasbay.com/api/v3`

4.1 Sign-in

FOR	SEND	NOTES
Servers, SDKs, plugins	`X-Api-Username: UID` `X-Api-Key: key`	From My Profile › API Access. Same rules as the classic API: own panel domain, allowed IPs.
Mobile and desktop apps	`Authorization: Bearer vb3_…`	Token from `POST /auth/login` (chapter 5); valid 90 days, revocable per device.

Credentials travel in headers only, never in the URL.

Postman: a collection of every v3 and classic endpoint, set up for this panel: `https://www.vasbay.com/api-documentation/postman.json`. Import it in Postman and fill the variables `uid`, `api_key`, `sender_id`, `mobile` and `schedule_time`.

API KEY

```
curl https://www.vasbay.com/api/v3/balance \
  -H 'X-Api-Username: YOUR_UID' \
  -H 'X-Api-Key: YOUR_API_KEY'
```

4.2 Endpoints

ENDPOINT	SIGN-IN	PARAMETERS / RETURNS
`POST /auth/login`	none	App sign-in: `login`, `password`, `device_name` → bearer token (90 days)
`POST /auth/logout`	token	Sign this device out

ENDPOINT	SIGN-IN	PARAMETERS / RETURNS
`GET /auth/devices` · `DELETE /auth/devices/{id}`	token	Signed-in devices of the account; sign one out
`GET /me`	token · key	Account: id, username, UID, name, e-mail, phone, company, account type, services
`GET /balance`	token · key	Credits: transactional, marketing, international (EUR)
`POST /sms/send`	token · key	`numbers` (array or comma list; strings or JSON numbers) and/or `group_ids` (contact groups, up to 50; `numbers` are then extra recipients), `message`, `sender_id`, `route`, `sms_type`, `schedule_time`, `remove_duplicate`, `template_id`, `entity_id`, `content_id` (send DLT ids as strings). A group send lists the first 100 numbers; `meta.count` is the full count
`POST /sms/bulk`	token · key	`recipients`: `[{mobile, ref, fields}]` with `message`, `sender_id`, `route`, `schedule_time`; `fields` fill `#FIELD2#` .... Personalised sends go out as one push per distinct text (`push_ids`, `push_id` is the first), at most 1000 recipients per request. A scheduled personalised send returns `schedule_id` (its `push_id` comes when sent)
`GET /sms/status`	token · key	`push_id`, `number_id` or `page`
`GET /sms/history`	token · key	`from`, `to` (YYYY-MM-DD), `page`, `via` (web, mobile, api), `scheduled`

ENDPOINT	SIGN-IN	PARAMETERS / RETURNS
`GET /sms/pushes/{push_id}/numbers`	token · key	Every number of one push, paged (`page` , `per_page` up to 100): mobile, status, `bucket` , error, sent and delivered time. `status` filter (`delivered` , `sent` , `failed` , `dnd` , `unknown` , `stopped`), `q` (part of a number); `meta.counts` has the totals of the whole push
`GET /groups`	token · key	Contact groups with their number of contacts; `q` (name), `page` , `per_page`
`GET /groups/{id}` · `GET /groups/{id}/contacts`	token · key	One group; its contacts paged (`q` matches number or name)
`GET /scheduled`	token · key	Scheduled SMS with status and allowed actions; `status` filter (also `failed`), `type` (`bulk` , `dynamic`)
`POST /scheduled/{push_id}/hold|resume|stop`	token · key	Hold, resume or stop a schedule (stop refunds unsent numbers)
`POST /scheduled/dynamic/{schedule_id}/stop`	token · key	Stop a personalised schedule before it is sent (nothing is charged yet)
`GET /sender-ids`	token · key	All sender IDs with status (`approved` , `under_review` , `with_operator` , `rejected`) and entity ID; filters `status` , `sender_id` , `route`
`GET /sender-ids/{id}`	token · key	Status of one sender ID request
`POST /sender-ids`	token · key	Request a sender ID: `sender_id` , `route` , `entity_id` , `company_name` , `explanation` , `purpose` , `sample_sms`

ENDPOINT	SIGN-IN	PARAMETERS / RETURNS
`GET /templates`	token · key	DLT templates with content ID, entity ID, sender ID and status; `type` (transactional, marketing), `variant` (static, dynamic), `status` (approved by default, or `all`...), `content_id`
`GET /templates/{id}`	token · key	Status of one template
`POST /templates`	token · key	Register a DLT template: `type`, `content_id`, `text`, `sender_id`, `entity_id` (defaults to the sender ID's), `variant`
`GET /notifications`	token · key	Notices for the account (HTML and plain text), newest first
`GET /`	none	API name and version

`schedule_time` accepts `2026-12-25 10:30` or ISO 8601; the send is queued and `numbers` in the reply stays empty until it goes out (chapter 9). `numbers` may be strings or JSON numbers; send `entity_id` and `content_id` as strings. Lists accept `page` and `per_page` (max 100). SMS sent with an app token show as Mobile in the panel reports, with an API key as API.

4.3 Examples and replies

SEND

```
curl -X POST https://www.vasbay.com/api/v3/sms/send \
  -H 'X-Api-Username: YOUR_UID' -H 'X-Api-Key: YOUR_API_KEY' \
  -H 'Content-Type: application/json' \
  -d '{"numbers": ["9895XXXXXX", "9746XXXXXX"], "message": "Dear Customer, Your SMS account has been
activated successfully.Texaavy Infotech",
       "sender_id": "TXAAVY", "route": 2}'
```

SUCCESS

```
HTTP/1.1 201 Created
{
  "success": true,
  "data": {
    "push_id": 14349171,
    "scheduled_at": null,
    "numbers": [
      { "id": 20222080, "mobile": "9895XXXXXX", "status": "Sent", "ref": "0" },
      { "id": 20222081, "mobile": "9746XXXXXX", "status": "Sent", "ref": "0" }
    ]
  },
  "meta": { "count": 2 }
}
```

ERROR

```
HTTP/1.1 422 Unprocessable Entity
{ "success": false, "error": { "code": "3076", "message": "Invalid Sender Id" } }
```

BULK / PERSONALISED

```
curl -X POST https://www.vasbay.com/api/v3/sms/bulk \
  -H 'X-Api-Username: YOUR_UID' -H 'X-Api-Key: YOUR_API_KEY' -H 'Content-Type: application/json' \
  -d '{"sender_id": "TXAAVY", "route": 2,
       "message": "Dear #FIELD2#, your bill of Rs.#FIELD3# is due. -Texaavy",
       "recipients": [
         {"mobile": "9895XXXXXX", "ref": "INV-1001", "fields": ["Dev", "1250"]},
         {"mobile": "9746XXXXXX", "ref": "INV-1002", "fields": ["Nithin", "980"]}
       ]}'
```

Scheduled bulk and dynamic: add `schedule_time` to `POST /sms/bulk`. Bulk replies with the `push_id` (hold / resume / stop under `/scheduled/{push_id}/…`). Dynamic (recipients with `fields`) replies with a `schedule_id` and `push_id: null`; the `push_id` appears in `GET /scheduled?type=dynamic` once sent, and `POST /scheduled/dynamic/{schedule_id}/stop` stops it until then. The balance is checked when the schedule is made and taken when it is sent.

SCHEDULED BULK

```
curl -X POST https://www.vasbay.com/api/v3/sms/bulk \
  -H 'X-Api-Username: YOUR_UID' -H 'X-Api-Key: YOUR_API_KEY' -H 'Content-Type: application/json' \
  -d '{"sender_id": "TXAAVY", "route": 2, "schedule_time": "2026-12-25 10:30",
       "message": "Dear Customer, Your SMS account has been activated successfully.Texaavy Infotech",
       "recipients": [
         {"mobile": "9895XXXXXX", "ref": "CUS-1"},
         {"mobile": "9746XXXXXX", "ref": "CUS-2"}
       ]}'
```

SCHEDULED DYNAMIC

```
curl -X POST https://www.vasbay.com/api/v3/sms/bulk \
  -H 'X-Api-Username: YOUR_UID' -H 'X-Api-Key: YOUR_API_KEY' -H 'Content-Type: application/json' \
  -d '{"sender_id": "TXAAVY", "route": 2, "schedule_time": "2026-12-25 10:30",
       "message": "Dear #FIELD2#, your bill of Rs.#FIELD3# is due. -Texaavy",
       "recipients": [
         {"mobile": "9895XXXXXX", "ref": "INV-1001", "fields": ["Dev", "1250"]},
         {"mobile": "9746XXXXXX", "ref": "INV-1002", "fields": ["Nithin", "980"]}
       ]}'
```

REPLY · DYNAMIC

```
HTTP/1.1 201 Created
{
  "success": true,
  "data": {
    "push_id": null,
    "scheduled_at": "2026-12-25T10:30:00+05:30",
    "numbers": [],
    "schedule_id": 42,
    "type": "dynamic"
  },
  "meta": { "count": 0 }
}
```

4.4 Sender IDs and DLT templates

Every SMS goes out under an approved sender ID and must match a DLT template registered with its content ID under your entity ID. Statuses: `under_review`, `with_operator`, `approved`, `rejected`.

`POST /sender-ids` requests a sender ID; `GET /sender-ids/{id}` shows its status and `GET /sender-ids` lists all (filters `status`, `sender_id`, `route`).

REQUEST A SENDER ID

```
curl -X POST https://www.vasbay.com/api/v3/sender-ids \
  -H 'X-Api-Username: YOUR_UID' -H 'X-Api-Key: YOUR_API_KEY' -H 'Content-Type: application/json' \
  -d '{"sender_id": "TXAAVY", "route": 2, "entity_id": "1701XXXXXXXXXXXXXX",
       "company_name": "Texaavy Infotech", "explanation": "TeXAAVY = Texaavy Infotech",
       "purpose": "OTP and order updates", "sample_sms": "Your OTP is 1234. -Texaavy"}'
```

REPLY

```
{
  "success": true,
  "data": {
    "id": 758, "sender_id": "TXAAVY", "route": 2, "route_name": "Transactional",
    "status": "under_review", "entity_id": "1701XXXXXXXXXXXXXX",
    "requested_at": "2026-10-24T10:15:00+05:30", "decided_at": null
  }
}
```

`POST /templates` registers a template: `type`, `content_id`, `text`, `sender_id` (one of yours on the same route) and `entity_id` (taken from the sender ID when left out, must match it when given). A content ID can be registered once; templates are approved at once, as in the panel. `GET /templates` lists approved templates (`status=all` for every status), `GET /templates/{id}` shows one.

REGISTER A TEMPLATE

```
curl -X POST https://www.vasbay.com/api/v3/templates \
  -H 'X-Api-Username: YOUR_UID' -H 'X-Api-Key: YOUR_API_KEY' -H 'Content-Type: application/json' \
  -d '{"type": "transactional", "content_id": "1707XXXXXXXXXXXXXX",
       "sender_id": "TXAAVY", "entity_id": "1701XXXXXXXXXXXXXX",
       "text": "Your OTP is {#var#}. -Texaavy"}'
```

REPLY

```
HTTP/1.1 201 Created
{
  "success": true,
  "data": {
    "id": 8130, "type": "transactional", "variant": "static", "route": 2, "status": "approved",
    "content_id": "1707XXXXXXXXXXXXXX", "entity_id": "1701XXXXXXXXXXXXXX", "sender_id": "TXAAVY",
    "text": "Your OTP is {#var#}. -Texaavy", "characters": 29, "sms_parts": 1, "variables": 1,
    "added_at": "2026-10-24T10:20:00+05:30", "decided_at": null
  }
}
```

4.4 HTTP status and limits

HTTP STATUS	MEANING
`200` / `201`	Success (201 for created: sign-in, send, sender ID request)
`401`	`601` wrong key, wrong or expired token, wrong domain or IP; `606` no credentials
`404` · `405`	`602` unknown endpoint or record; `405` wrong method
`422`	Validation: `3xxx` codes of the error list (`3087` names the invalid field)
`413` · `429`	`607` body over 20 MB; `429` more than 240 requests a minute (login: 20 a minute)
`500`	Unexpected error; retry later (never retry a send blindly: check status first)

Every reply carries `X-RateLimit-Limit` and `X-RateLimit-Remaining`. Error codes are those of chapter 13.

5 Mobile apps

Apps never contain an API key. Each user signs in with their own panel login and the app receives a token for that device only.

SIGN-IN FLOW

```
curl -X POST https://www.vasbay.com/api/v3/auth/login -H 'Content-Type: application/json' \
  -d '{"login": "your-login", "password": "your-password", "device_name": "Pixel 8"}'

# reply (HTTP 201): keep the token in the device's secure storage
{ "success": true, "data": { "token": "vb3_…", "token_type": "Bearer",
  "expires_at": "2027-01-18T10:30:00+05:30", "user": { "id": 1024, "username": "…", "account_type":
"customer" } } }

# every later call
curl https://www.vasbay.com/api/v3/me -H 'Authorization: Bearer vb3_…'
```

- **Store the token securely:** iOS Keychain, Android Keystore / EncryptedSharedPreferences; never in plain preferences or logs.
- **On HTTP 401** (expired or signed out) show the sign-in screen again; tokens last 90 days.

- **Sign-in rules** are those of the panel: the account must belong to this panel's domain; 8 wrong passwords lock the login for 15 minutes; at most 20 sign-in requests a minute per address.
- **Devices:** `GET /auth/devices` lists the signed-in devices (up to 10, the oldest is signed out first), `DELETE /auth/devices/{id}` signs one out, `POST /auth/logout` the current one.
- Everything else is the same API as for servers (chapter 4).

6 SDKs and plugins

Ready-made integrations on the v3 API, prepared for this panel. They contain no credentials: each customer enters their own Username (UID) and API key from My Profile › API Access. Download them from the online documentation (https://www.vasbay.com/api-documentation#sdks).

PACKAGE	PLATFORM	VERSION	DOWNLOAD
JavaScript SDK	Node.js 18+	3.1.0	https://www.vasbay.com/integrations/javascript.zip SHA-256 f9c6afac56a1f4af3d1d53d2…
PHP SDK	PHP 7.4+	3.1.0	https://www.vasbay.com/integrations/php.zip SHA-256 bafe2f24ca2f0b02a0a2ddf4…
Python SDK	Python 3.8+	3.1.0	https://www.vasbay.com/integrations/python.zip SHA-256 788437e16499d0abc80170b5…
Laravel package	Laravel 9–12, PHP 8.1+	3.1.0	https://www.vasbay.com/integrations/laravel.zip SHA-256 51589f0ad3cc0b5fb2d595de…
WordPress plugin	WordPress 5.8+, WooCommerce	3.1.0	https://www.vasbay.com/integrations/wordpress.zip SHA-256 737f01d0e33eb5fdac342ad1…
Magento 2 module	Magento 2.4	3.1.0	https://www.vasbay.com/integrations/magento.zip SHA-256 bffcbbc267ef44457e8a323e…

JAVASCRIPT (NODE.JS 18+, SERVER ONLY)

```
// npm install ./sms-gateway-js            (Node.js 18+, server only)
import { SmsGateway } from '@texaavy/sms-gateway';

const sms = new SmsGateway({ username: process.env.SMS_USERNAME, apiKey: process.env.SMS_API_KEY });
const r = await sms.send({ numbers: ['9895XXXXXX'], message: 'Dear Customer, Your SMS account has been
activated successfully.Texaavy Infotech', senderId: 'TXAAVY' });
console.log(r.push_id, await sms.balance());
```

PHP

```
// require 'texaavy/sms-gateway' (path repository) or include src/Client.php
use Texaavy\SmsGateway\Client;

$sms = new Client(['username' => getenv('SMS_USERNAME'), 'apiKey' => getenv('SMS_API_KEY')]);
$r = $sms->send(['numbers' => ['9895XXXXXX'], 'message' => 'Dear Customer, Your SMS account has been
activated successfully.Texaavy Infotech', 'senderId' => 'TXAAVY']);
echo $r['push_id'];
```

PYTHON

```
# pip install ./sms-gateway-python
import os
from sms_gateway import SmsGateway

sms = SmsGateway(username=os.environ["SMS_USERNAME"], api_key=os.environ["SMS_API_KEY"])
r = sms.send(numbers=["9895XXXXXX"], message="Dear Customer, Your SMS account has been activated
successfully.Texaavy Infotech", sender_id="TXAAVY")
print(r["push_id"], sms.balance())
```

LARAVEL

```
// .env
SMS_GATEWAY_USERNAME=YOUR_UID
SMS_GATEWAY_API_KEY=YOUR_API_KEY
SMS_GATEWAY_SENDER_ID=TXAAVY

// anywhere
SmsGateway::send(['numbers' => ['9895XXXXXX'], 'message' => 'Dear Customer, Your SMS account has been
activated successfully.Texaavy Infotech']);

// notifications: via() returns ['sms-gateway'], toSmsGateway() returns the text
public function toSmsGateway($notifiable)
{
    return SmsGatewayMessage::create('Dear Customer, your order is shipped. -Texaavy');
}
```

WORDPRESS / WOOCOMMERCE

```
1. Plugins › Add New › Upload Plugin › choose sms-gateway-wordpress-3.0.0.zip › Activate
2. Settings › SMS Gateway: Username (UID), API key, sender ID and route › Save settings
3. Optional: tick the WooCommerce order statuses that send an SMS and edit each text
   (it must match an approved DLT template). Placeholders:
   {customer_name} {order_number} {order_total} {order_status} {store_name}
```

MAGENTO 2

```
# from the Magento root
unzip sms-gateway-magento-3.0.0.zip -d .        # creates app/code/Texaavy/SmsGateway
bin/magento module:enable Texaavy_SmsGateway
bin/magento setup:upgrade && bin/magento cache:flush

# Stores › Configuration › Texaavy › SMS Gateway: Username (UID), API key, sender ID,
# then "Test connection" and the order placed / order shipped SMS texts
```

PHP SDK with Composer

The official VASbay PHP SDK can be installed with Composer instead of the zip: SMS, voice calls, balance and account, sender IDs, DLT templates, contact groups and scheduled SMS, in any PHP project or in Laravel.

Package	texaavy/vasbay-php-sdk
Source	https://github.com/Texaavyinfotech/vasbay-php-sdk
Install	composer require texaavy/vasbay-php-sdk:dev-main
Requires	PHP 7.4+ with ext-curl and ext-json; Laravel optional (auto-discovered)
Licence	MIT

INSTALL

```
# in your project folder (PHP 7.4+ with ext-curl and ext-json)

# 1. add the GitHub repository (needed until the package is listed on Packagist)
composer config repositories.vasbay vcs https://github.com/Texaavyinfotech/vasbay-php-sdk

# 2. install the package
composer require texaavy/vasbay-php-sdk:dev-main

# 3. keep the credentials out of the code (My Profile › API Access)
export VASBAY_USERNAME="your-uid"
export VASBAY_API_KEY="your-api-key"
```

PHP

```
require __DIR__ . '/vendor/autoload.php';

use Texaavy\Vasbay\Client;
use Texaavy\Vasbay\Exceptions\VasbayException;

$vasbay = new Client(getenv('VASBAY_USERNAME'), getenv('VASBAY_API_KEY'));

try {
    $sent = $vasbay->sms()->send(['to' => '9895XXXXXX', 'message' => 'Dear Customer, Your SMS account
has been activated successfully.Texaavy Infotech', 'senderId' => 'TXAAVY']);
    echo 'push_id ', $sent['push_id'], PHP_EOL;

    $status  = $vasbay->sms()->status($sent['push_id']);      // delivery status
    $balance = $vasbay->account()->balance();                 // transactional, marketing,
international_eur
    echo 'Transactional credits: ', $balance['transactional'], PHP_EOL;
} catch (VasbayException $e) {
    echo $e->getErrorCode(), ' ', $e->getMessage();          // e.g. 3076 Invalid Sender Id
}

// also: sms()->sendBulk(), sms()->history(), voice()->send(), senderIds(), templates(), scheduled(),
groups()
```

LARAVEL

```
// composer require texaavy/vasbay-php-sdk:dev-main   (service provider and facade are auto-discovered)

// .env
VASBAY_USERNAME=YOUR_UID
VASBAY_API_KEY=YOUR_API_KEY
VASBAY_SENDER_ID=TXAAVY

// anywhere
use Texaavy\Vasbay\Laravel\Facades\Vasbay;

Vasbay::sms()->send(['to' => '9895XXXXXX', 'message' => 'Dear Customer, Your SMS account has been
activated successfully.Texaavy Infotech', 'senderId' => 'TXAAVY']);
Vasbay::account()->balance();

// notifications: via() returns ['vasbay'], toVasbay() returns the message
public function toVasbay($notifiable)
{
    return \Texaavy\Vasbay\Laravel\VasbayMessage::create('Dear Customer, your order is shipped. -
Texaavy');
}
```

dev-main installs the latest code of the main branch. Until the package is listed on Packagist, run the "composer config repositories.vasbay vcs ..." line first. Full guide: https://github.com/Texaavyinfotech/vasbay-php-sdk

Security built in: HTTPS with certificate checks, credentials in headers and stored encrypted (WordPress, Magento) or in environment variables, keys never logged or shown again, no automatic retry of sends (no duplicate SMS), admin screens protected by permissions and CSRF tokens. The JavaScript SDK refuses to run in a browser. Versions and checksums: https://www.vasbay.com/integrations/manifest.json

7 Send SMS (classic API)

GET / POST `https://www.vasbay.com/webservice/smsSimpleApi.php`

Sends one message to one or more numbers. Accepts every input style; the numbers go in `numbers`.

7.1 Parameters

PARAMETER		DESCRIPTION
`usersName`	**required**	Your UID (hex)
`key`	**required**	API key
`route`	**required**	`1` promotional (marketing) · `2` transactional · `4` international
`senderId`	**routes 1, 2**	An approved sender id (header) of your account, case-sensitive, e.g. `TXAAVY`
`message`	**required**	The text. Must match one of your approved DLT templates unless your account has an open route. URL-encode it in a URL.
`numbers`	**required**	Mobile numbers separated by commas (JSON: also an array). 10 digits; a leading `0` or `+` is removed.
`smsType`	optional	`1` normal (default) · `2` flash · `3` unicode (any language)
`removeDuplicate`	optional	`1` send once to a repeated number · `0` (default) send to every entry
`creditBack`	optional	`1` normal (default) · `2` credit back, where your account has it enabled
`scheduleType`	optional	`1` send now (default) · `2` send once at `scheduleTime`. Repeating schedules are made in the panel.

PARAMETER		DESCRIPTION
`scheduleTime`	with scheduleType 2	`DD-MM-YYYY hh:mm AM`, Indian time, in the future, e.g. `25-12-2026 10:30 AM`; in a URL encode the spaces: `25-12-2026%2010:30%20AM`
`smsTemplateId`	optional	Id of the approved template to use; found automatically from the message when left out
`entityId`, `contentId`	optional	DLT entity id and content (template) id, numeric; looked up from the template when left out
`includeNumber`, `excludeNumber`	optional	Extra numbers to add / numbers to leave out, comma separated
`format`	optional	`json` (default) · `xml` · `soap`

7.2 Examples

URL (GET)

```
https://www.vasbay.com/webservice/smsSimpleApi.php?usersName=YOUR_UID&key=YOUR_API_KEY&route=2&senderId
=TXAAVY&numbers=9895XXXXXX,9746XXXXXX&message=Dear%20Customer%2C%20Your%20SMS%20account%20has%20been%20
activated%20successfully.Texaavy%20Infotech
```

FORM POST (CURL)

```
curl -X POST 'https://www.vasbay.com/webservice/smsSimpleApi.php' \
  -d usersName=YOUR_UID -d key=YOUR_API_KEY \
  -d route=2 -d senderId=TXAAVY -d numbers=9895XXXXXX \
  --data-urlencode 'message=Dear Customer, Your SMS account has been activated successfully.Texaavy
Infotech'
```

JSON BODY

```
curl -X POST 'https://www.vasbay.com/webservice/smsSimpleApi.php' \
  -H 'Content-Type: application/json' \
  -d '{"usersName":"YOUR_UID","key":"YOUR_API_KEY","route":"2","senderId":"TXAAVY",
      "numbers":["9895XXXXXX","9746XXXXXX"],"message":"Dear Customer, Your SMS account has been
activated successfully.Texaavy Infotech"}'
```

XML RESPONSE REQUESTED

```
https://www.vasbay.com/webservice/smsSimpleApi.php?usersName=YOUR_UID&key=YOUR_API_KEY&route=2&senderId=TXAAVY&numbers=9895XXXXXX&message=Dear%20Customer%2C%20Your%20SMS%20account%20has%20been%20activated%20successfully.Texaavy%20Infotech&format=xml
```

7.3 Response

FIELD	DESCRIPTION
`pushId`	Identifies the whole send; use it for delivery status
`smsHistoryNumberId`	Identifies one number of the send
`apirefSmsId`	Your own reference (`idno` of a bulk send), otherwise `0`
`status`	Status at the time of the answer (see chapter 11)
`count`	Numbers listed in this answer

The list holds the numbers created so far; large sends are queued and appear in delivery status as they are processed.

200 OK · JSON

```
{
  "data": "0",
  "msg": {
    "pushId": 14349171,
    "response": {
      "0": { "smsHistoryNumberId": 20222080, "status": "Sent", "apirefSmsId": "30", "phone": 9895XXXXXX
},
      "1": { "smsHistoryNumberId": 20222081, "status": "Sent", "apirefSmsId": "31", "phone": 9746XXXXXX
},
      "count": 2
    }
  }
}
```

200 OK · XML

```
<?xml version="1.0"?>
<result>
  <data>0</data>
  <msg>
    <pushId>14349171</pushId>
    <response>

<n0><smsHistoryNumberId>20222080</smsHistoryNumberId><status>Sent</status><apirefSmsId>30</apirefSmsId>
<phone>9895XXXXXX</phone></n0>

<n1><smsHistoryNumberId>20222081</smsHistoryNumberId><status>Sent</status><apirefSmsId>31</apirefSmsId>
<phone>9746XXXXXX</phone></n1>
      <count>2</count>
    </response>
  </msg>
</result>
```

8 Bulk and dynamic SMS

POST `https://www.vasbay.com/webservice/smsApi.php?usersName=…&key=…`

Sends the same message to many numbers with your own reference per number (`idno` , returned as `apirefSmsId`), or a personalised message per number. The body is JSON or XML; the head takes the parameters of chapter 7 except `numbers` . The response is the same as in section 7.3.

ENTRY FIELD		DESCRIPTION
`mobile`	static bulk	The number. Entries without a valid 10-digit number are skipped.
`idno`	optional	Your reference for this number, returned as `apirefSmsId`
`field1` … `fieldN`	dynamic	`field1` is the number, `field2` … replace `#FIELD2#` … in the message

JSON · STATIC BULK

```
curl -X POST 'https://www.vasbay.com/webservice/smsApi.php?usersName=YOUR_UID&key=YOUR_API_KEY' \
  -H 'Content-Type: application/json' \
  -d '{
  "smsRequest": {
    "head": {
      "route": "2",
      "senderId": "TXAAVY",
      "smsType": "1",
      "removeDuplicate": "1",
      "message": "Dear Customer, Your SMS account has been activated successfully.Texaavy Infotech"
    },
    "request": [
      { "mobile": "9895XXXXXX", "idno": "30" },
      { "mobile": "9746XXXXXX", "idno": "31" }
    ]
  }
}'
```

XML · STATIC BULK

```
<?xml version="1.0" encoding="UTF-8"?>
<smsRequest>
  <head>
    <route>2</route>
    <senderId>TXAAVY</senderId>
    <smsType>1</smsType>
    <removeDuplicate>1</removeDuplicate>
    <message>Dear Customer, Your SMS account has been activated successfully.Texaavy Infotech</message>
  </head>
  <request><mobile>9895XXXXXX</mobile><idno>30</idno></request>
  <request><mobile>9746XXXXXX</mobile><idno>31</idno></request>
</smsRequest>
```

XML · DYNAMIC (PERSONALISED)

```
<?xml version="1.0" encoding="UTF-8"?>
<smsRequest>
  <head>
    <route>2</route>
    <senderId>TXAAVY</senderId>
    <message>Dear #FIELD2#, your bill of Rs.#FIELD3# is due on #FIELD4#. -Texaavy</message>
  </head>

<request><idno>33</idno><field1>9895XXXXXX</field1><field2>Dev</field2><field3>1250</field3><field4>25-10</field4></request>

<request><idno>34</idno><field1>9746XXXXXX</field1><field2>Nithin</field2><field3>980</field3><field4>27-10</field4></request>
</smsRequest>
```

POSTING AN XML FILE

```
curl -X POST
'https://www.vasbay.com/webservice/smsApi.php?usersName=YOUR_UID&key=YOUR_API_KEY&format=xml' \
  -H 'Content-Type: application/xml' \
  --data-binary @request.xml
```

9 Scheduled SMS

Add `scheduleType=2` and `scheduleTime` (`DD-MM-YYYY hh:mm AM/PM`, Indian time) to any send. The time must be in the future; in a URL encode its spaces (`%20`). Promotional SMS (route 1) are only sent between 9:10 AM and 8:45 PM, so a promotional schedule must fall in that window too.

A scheduled send is queued, not sent: the reply carries the `pushId` and the time, with no numbers yet. The numbers and their status appear under `smsStatusApi.php?pushId=` once the time has come.

Bulk and dynamic: request entries (`mobile`, or `field1`… for personalised messages, chapter 8) are scheduled the same way. A personalised schedule replies with a `scheduleId` and `pushId` `0`: its `pushId` is created when it is sent. List it with `GET /api/v3/scheduled?type=dynamic` and stop it with `POST /api/v3/scheduled/dynamic/{scheduleId}/stop`. Repeating schedules are made in the panel.

REQUEST

```
https://www.vasbay.com/webservice/smsSimpleApi.php?usersName=YOUR_UID&key=YOUR_API_KEY&route=2&senderId=TXAAVY&numbers=9895XXXXXX&message=Dear%20Customer%2C%20Your%20SMS%20account%20has%20been%20activated%20successfully.Texaavy%20Infotech&scheduleType=2&scheduleTime=25-12-2026%2010%3A30%20AM
```

REPLY

```
{
  "data": "0",
  "msg": {
    "pushId": 14349174,
    "scheduleTime": "25 December 2026 10:30 AM",
    "response": { "count": 0 }
  }
}
```

Your balance is checked when the schedule is created; the credits are taken when it is sent. Scheduled sends can be followed, put on hold, resumed or stopped in the panel under **Delivery Reports › Scheduled SMS**; stopping returns the credits of the numbers not yet sent. SMS sent through the API are marked **API** in the panel reports.

10 Credit balance

GET / POST `https://www.vasbay.com/webservice/smsBalanceApi.php`

Parameters: `usersName`, `key` and optionally `format`. Credits are returned as strings.

REQUEST

```
https://www.vasbay.com/webservice/smsBalanceApi.php?usersName=YOUR_UID&key=YOUR_API_KEY&format=json
```

200 OK · JSON

```
{
  "status": "0",
  "response": {
    "transactional": "20000",
    "marketing": "30000",
    "international": { "eur": "0" }
  }
}
```

200 OK · XML

```
<?xml version="1.0"?>
<result>
  <status>0</status>
  <response>
    <transactional>20000</transactional><marketing>30000</marketing>
    <international><eur>0</eur></international>
  </response>
</result>
```

11 Delivery status

GET / POST `https://www.vasbay.com/webservice/smsStatusApi.php`

PARAMETER		DESCRIPTION
`usersName`, `key`	**required**	Credentials
`pushId`	**one of**	Every number of one send
`smsHistoryNumberId`	**one of**	One number (with or without `pushId`)
`page`	**one of**	Your latest numbers, 10 per page (1, 2, ...); combine with `pushId` to page through a large send
`format`	optional	`json` · `xml` · `soap`

BY PUSH

```
https://www.vasbay.com/webservice/smsStatusApi.php?usersName=YOUR_UID&key=YOUR_API_KEY&format=json&pushId=14349171
```

ONE NUMBER

```
https://www.vasbay.com/webservice/smsStatusApi.php?usersName=YOUR_UID&key=YOUR_API_KEY&format=json&smsHistoryNumberId=20222080
```

PAGED (10 PER PAGE)

```
https://www.vasbay.com/webservice/smsStatusApi.php?usersName=YOUR_UID&key=YOUR_API_KEY&format=json&page=1
```

200 OK · JSON


```
{
  "data": "0",
  "msg": {
    "0": {
      "smsHistoryNumberId": 20222080,
      "smsHistoryId": 14349171,
      "phone": 9895XXXXXX,
      "status": "Delivered",
      "apirefSmsId": "30",
      "countSms": 2
    },
    "count": 1,
    "totalCount": 2
  }
}
```


`count` is the number of entries in this answer, `totalCount` the numbers of the push and `countSms` the numbers of the entry's push.

11.1 Status values

STATUS	MEANING
`Sent` / `Submitted`	Handed to the operator, delivery report not received yet
`Delivered`	Delivered to the handset
`Not Delivered`, `Failed`, operator codes	Not delivered (switched off, invalid number, ...)
`DND`, `Message Rejected`	Blocked by DND preference or rejected by the operator / DLT scrubbing
`Stopped`	The send was stopped in the panel before this number went out; it was not charged
`0` or empty	Pending – no status from the operator yet

Delivery reports usually arrive within minutes, but some operators take much longer. Poll sparingly, for example every few minutes per push.

12 SOAP API

POST https://www.vasbay.com/webservice/soap.php

GET https://www.vasbay.com/webservice/soap.php?wsdl

SOAP 1.1, document/literal, namespace `urn:vasbay:sms`. The operation is the first element of the Body and its children are the same fields as the REST API; namespace prefixes are optional. Repeat `<request>` for bulk or dynamic recipients. The answer is `<{operation}Response><result>…</result>` with the same fields as the REST JSON; add `format=json` or `format=xml` to the URL to receive those instead.

OPERATION	FIELDS
`sendSms`	as Send SMS / Bulk & dynamic SMS
`getBalance`	`usersName`, `key`
`getStatus`	as Delivery status

SENDSMS

```
POST /webservice/soap.php HTTP/1.1
Host: www.vasbay.com
Content-Type: text/xml; charset=utf-8
SOAPAction: "urn:vasbay:sms#sendSms"

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sms="urn:vasbay:sms">
  <soap:Body>
    <sms:sendSms>
      <sms:usersName>YOUR_UID</sms:usersName>
      <sms:key>YOUR_API_KEY</sms:key>
      <sms:route>2</sms:route>
      <sms:senderId>TXAAVY</sms:senderId>
      <sms:message>Dear Customer, Your SMS account has been activated successfully.Texaavy
Infotech</sms:message>
      <sms:request><sms:mobile>9895XXXXXX</sms:mobile><sms:idno>30</sms:idno></sms:request>
      <sms:request><sms:mobile>9746XXXXXX</sms:mobile><sms:idno>31</sms:idno></sms:request>
    </sms:sendSms>
  </soap:Body>
</soap:Envelope>
```

GETBALANCE

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sms="urn:vasbay:sms">
  <soap:Body>
    <sms:getBalance>
      <sms:usersName>YOUR_UID</sms:usersName>
      <sms:key>YOUR_API_KEY</sms:key>
    </sms:getBalance>
  </soap:Body>
</soap:Envelope>
```

GETSTATUS

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sms="urn:vasbay:sms">
  <soap:Body>
    <sms:getStatus>
      <sms:usersName>YOUR_UID</sms:usersName>
      <sms:key>YOUR_API_KEY</sms:key>
      <sms:pushId>14349171</sms:pushId>
    </sms:getStatus>
  </soap:Body>
</soap:Envelope>
```

RESPONSE

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getBalanceResponse xmlns="urn:vasbay:sms">
      <result>
        <status>0</status>

<response><transactional>20000</transactional><marketing>30000</marketing><international><eur>0</eur></
international></response>
      </result>
    </getBalanceResponse>
  </soap:Body>
</soap:Envelope>
```

Errors come back inside the same response element, for example `<result><status>601</status><response>Authentication Failed</response></result>` (HTTP 200, no SOAP Fault).

13 Error codes

`0` means success; any other code is an error. The message column is the exact text the API returns.

1. Request

CODE	MESSAGE	WHEN
601	Authentication Failed	usersName / key wrong, key disabled, account inactive, request not on your panel domain, or caller IP not allowed
602	Error Request Api Not Found	unknown SOAP operation
603	Invalid Response Format Supported Formats Are json, xml, soap	format is not json, xml or soap
604	Error No Contacts Found	no valid mobile number in numbers / request entries
605	Invalid Request Body Send Url Parameters, json, xml Or soap	the body is not valid JSON, XML or SOAP (or contains a DOCTYPE)
606	usersName And key Are Required	usersName or key missing
607	Request Too Large (Max 20 MB)	request body larger than 20 MB
500	Internal Error Please Try Again Later	unexpected server error; retry later

2. Send SMS

CODE	MESSAGE	WHEN
3001	Invalid Sms Route	route missing or not 1 / 2 / 4
3002	Invalid Sms Credit Back	creditBack is not 1 or 2
3003	Invalid Sms Type	smsType is not 1, 2 or 3
3004	Invalid Shedule Type	scheduleType is not 1 or 2 (repeating schedules are made in the panel)

CODE	MESSAGE	WHEN
3006	Number Must be Numeric	a number is not numeric
3007	Number Must be Have 10 Digit	a number does not have 10 digits
3012	Message Feild is Empty	message empty
3013	Sms Service Deactivated Please Contact Support To Activate Service	SMS service not active on the account
3014	Operator Downtime...Please Try Again Later	route under operator downtime
3017	Please Add Ateast One Mobile Number to Proceed	no number left to send to
3018	Your sms Credit Balance is Low Please Credit Sms	not enough credits on the route (a schedule is checked when it is created)
3019	Sms DownTIme Please Try After Some Time	route temporarily unavailable
3021	Invalid Sender Id	sender id not approved for the account
3022	Invalid Time Format	scheduleTime not readable
3030	Schedule Not Possible In Between 8:45 PM AND 9:10 AM	promotional SMS (now or scheduled) outside 9:10 AM – 8:45 PM
3031	Schedule Date And Time Must Greather Than Current Time (IST)	scheduleTime is in the past
3036	Empty SenderID	senderId missing (routes 1 and 2)
3075	Remove Duplicate Must 1 or 0	removeDuplicate is not 0 or 1
3076	Invalid Sender Id	senderId is not one of your approved sender ids
3077	Requested Template Not Approved Please Request Template From Panel	no approved template on the account (accounts without open route)
3080	Please Select valid template	smsTemplateId is not an approved template

CODE	MESSAGE	WHEN
3081	please select a template for Transaction sms	transactional SMS without a template
3082	Selected template Not Match with the enter template	message does not match any approved template
3084	scheduleTime Is Required For Scheduled Sms (Eg 22-03-2018 11:12 PM)	scheduleType 2 without scheduleTime
3087	Invalid {parameter}	a parameter has a value that is not allowed (the message names it), message is longer than 2000 characters
3088	Too Many Recipients In One Request (Max 100000)	more than 100000 request entries in one call
3301	Invalid Entity Id	entityId not numeric
3302	Invalid Content Id	contentId not numeric

3. Sender IDs & templates (v3)

CODE	MESSAGE	WHEN
3036	The sender ID must be exactly 6 letters or digits (for example TXAAVY).	sender ID request incomplete or invalid (the message names the field), or already requested on that route
3076	Sender ID TXAAVY is not one of your sender IDs on the transactional route (requested or approved).	sender_id of a template is not one of your sender IDs on its route
3301	entity_id does not match the DLT entity ID of sender ID TXAAVY (1701…).	entity_id of a template differs from its sender ID's DLT entity ID

CODE	MESSAGE	WHEN
3087	Content ID 1707... is already saved as template #8130.	template field missing or invalid, dynamic template without #FIELD#, or content ID already saved
602	Template not found	GET /sender-ids/{id} or /templates/{id} of an ID that is not yours

4. Status

CODE	MESSAGE	WHEN
3006	smsHistoryNumberId Must be Numeric	smsHistoryNumberId not numeric
3032	Invalid Push Id	pushId not numeric or not one of your pushes
3078	Please Provide pushId or smsHistoryNumberId to get Status	neither pushId, smsHistoryNumberId nor page given
3085	Invalid page Number	page is not a positive number
3086	No Sms Found	no SMS found for the given smsHistoryNumberId / page

14 Code samples

Each sample sends one SMS with a form POST and reads the JSON answer. Replace `YOUR_UID` and `YOUR_API_KEY` with your credentials.

PHP (CURL)

```
<?php
// Send an SMS (JSON response)
$query = http_build_query([
    'usersName' => 'YOUR_UID',
    'key'       => 'YOUR_API_KEY',
    'route'     => 2,
    'senderId'  => 'TXAAVY',
    'numbers'   => '9895XXXXXX,9746XXXXXX',
    'message'   => 'Dear Customer, Your SMS account has been activated successfully.Texaavy Infotech',
]);
$ch = curl_init('https://www.vasbay.com/webservice/smsSimpleApi.php');
curl_setopt_array($ch, [
    CURLOPT_POST           => true,
    CURLOPT_POSTFIELDS     => $query,
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_TIMEOUT        => 30,
]);
$result = json_decode(curl_exec($ch), true);
curl_close($ch);

if (($result['data'] ?? null) === '0') {
    echo 'Accepted, push id ' . $result['msg']['pushId'];
} else {
    echo 'Error ' . $result['status'] . ': ' . $result['response'];
}
```

PYTHON (REQUESTS)

```
import requests

r = requests.post("https://www.vasbay.com/webservice/smsSimpleApi.php", data={
    "usersName": "YOUR_UID",
    "key": "YOUR_API_KEY",
    "route": 2,
    "senderId": "TXAAVY",
    "numbers": "9895XXXXXX,9746XXXXXX",
    "message": "Dear Customer, Your SMS account has been activated successfully.Texaavy Infotech",
}, timeout=30)
res = r.json()
if res.get("data") == "0":
    print("Accepted, push id", res["msg"]["pushId"])
else:
    print("Error", res["status"], res["response"])
```

NODE.JS (18+, FETCH)

```
const body = new URLSearchParams({
  usersName: "YOUR_UID",
  key: "YOUR_API_KEY",
  route: "2",
  senderId: "TXAAVY",
  numbers: "9895XXXXXX,9746XXXXXX",
  message: "Dear Customer, Your SMS account has been activated successfully.Texaavy Infotech",
});
const res = await (await fetch("https://www.vasbay.com/webservice/smsSimpleApi.php", { method: "POST",
body })).json();
if (res.data === "0") console.log("Accepted, push id", res.msg.pushId);
else console.log("Error", res.status, res.response);
```

JAVA (11+, HTTPCLIENT)

```
HttpClient client = HttpClient.newHttpClient();
String form = "usersName=YOUR_UID&key=YOUR_API_KEY&route=2&senderId=TXAAVY&numbers=9895XXXXXX"
            + "&message=" + URLEncoder.encode("Dear Customer, Your SMS account has been activated
successfully.Texaavy Infotech", StandardCharsets.UTF_8);
HttpRequest req =
HttpRequest.newBuilder(URI.create("https://www.vasbay.com/webservice/smsSimpleApi.php"))
    .header("Content-Type", "application/x-www-form-urlencoded")
    .POST(HttpRequest.BodyPublishers.ofString(form))
    .build();
String json = client.send(req, HttpResponse.BodyHandlers.ofString()).body();
```

15 Notes and limits

TOPIC	DETAILS
Templates (DLT)	Transactional and promotional messages must match an approved template of your account, including the sender id. Request templates and sender ids in the panel.
Message length	160 characters per SMS part (153 when split into several parts); unicode 70 (67 when split). Credits are charged per part and number.
Credits	Checked before sending; a send larger than your balance is refused with `3018`.
Timeouts	Allow at least 30 seconds for a send. Do not retry a send automatically on a timeout without checking delivery status first, or the SMS may go out twice.
Delivery reports	Usually within minutes, but some operators take much longer. Poll `smsStatusApi.php` sparingly.
Numbers	10-digit mobile numbers; a leading `0` or `+` is removed. Invalid entries of a bulk send are skipped.
Security	Use HTTPS, keep the key on your server (never in a mobile app or web page) and restrict it to your server IP addresses.

A Appendix: quick reference

TASK	CALL	REQUIRED
Send SMS	`smsSimpleApi.php` (GET/POST)	usersName, key, route, senderId, numbers, message
Bulk / dynamic SMS	`smsApi.php` (POST JSON/XML)	usersName, key, head: route, senderId, message; request entries
Schedule SMS	any send + `scheduleType=2`	scheduleTime `DD-MM-YYYY hh:mm AM`
Credit balance	`smsBalanceApi.php`	usersName, key

TASK	CALL	REQUIRED
Delivery status	`smsStatusApi.php`	usersName, key and pushId, smsHistoryNumberId or page
SOAP	`soap.php` · WSDL `soap.php?wsdl`	operation sendSms / getBalance / getStatus
Response format	`format=`	json (default) · xml · soap
API v3	`https://www.vasbay.com/api/v3`	headers X-Api-Username + X-Api-Key, or Authorization: Bearer (apps)
App sign-in	`POST /api/v3/auth/login`	login, password, device_name

B Appendix: SMPP connectivity

High-volume senders can connect over SMPP v3.4 instead of HTTP. SMPP messages are sent under your VASbay account like API sends: credits, approved sender IDs and DLT templates apply and every message appears in your delivery reports. Host, port and credentials are shared securely with each SMPP account (one account per route).

ITEM	VALUE
Binds	Transceiver (recommended), transmitter, receiver · interface version 0x34 · IP whitelisted · TPS and bind count per account
`source_addr`	Approved sender ID (TON 5 / NPI 0)
`destination_addr`	Number with country code, e.g. 919876543210
`data_coding`	0 default text, 1 ASCII, 3 Latin-1, 8 UCS-2 (Unicode); 0x10 / 0x18 flash
Long SMS	UDH (esm_class 0x40, 8 / 16-bit reference) or `message_payload` 0x0424; each part has its own message_id and receipt
DLT TLVs	0x1400 entity ID · 0x1401 template ID · 0x1402 telemarketer ID (optional)
`registered_delivery`	1 always · 2 failures only · 0 none
Receipts	`deliver_sm` esm_class 0x04 on a receiver / transceiver bind: `id: sub: dlvrd: submit date: done date: stat: err:`, TLVs 0x001E receipted_message_id and 0x0427 message_state. stat DELIVRD, UNDELIV, EXPIRED, REJECTD (err = API error, e.g. 076 = 3076)
Other PDUs	`enquire_link` (every 30–60 s), `query_sm`, `unbind`

Command status codes

STATUS	NAME	WHEN
0x0E	ESME_RINVPASWD	Wrong password
0x0F	ESME_RINVSYSID	Unknown or disabled system ID
0x0D	ESME_RBINDFAIL	IP not whitelisted or bind limit reached

STATUS	NAME	WHEN
0x04	ESME_RINVBNDSTS	submit_sm on a receiver bind
0x58	ESME_RTHROTTLED	Above your TPS – retry shortly
0x14	ESME_RMSGQFUL	Too many queued messages
0x0A / 0x0B	ESME_RINVSRCADR / RINVDSTADR	Invalid sender / number
0x104	ESME_RINVDCS	Binary data coding
0x08	ESME_RSYSERR	Temporary error – retry

VASbay SMS API v3.0 · 4 Nov 2026 · the online version at https://www.vasbay.com/api-documentation is always the most recent.