



Povikvane

Povikvane Public API

Ръководство за интеграция за клиенти

SMS и Viber съобщения · Версия 1.3



SHTRAK
WE ARE ON A MISSION

Всички права запазени

Съдържание

- 1 [Общ преглед](#)
- 2 [Първи стъпки](#)
- 3 [Удостоверяване](#)
- 4 [Изпращане на съобщение](#)
- 5 [Параметри на заявката](#)
- 6 [Идемпотентност \(безопасни повторни опити\)](#)
- 7 [Успешен отговор](#)
- 8 [Проверка на статуса на съобщение](#)
- 9 [Статуси и жизнен цикъл на съобщенията](#)
- 10 [Уебхукове \(push известия\)](#)
- 11 [Лимити на заявките](#)
- 12 [Кредитна система](#)
- 13 [Справочник на грешките](#)
- 14 [Правила за валидация](#)
- 15 [Примери с код](#)
- 16 [Поддръжка](#)

1 Общ преглед

Публичното API на Povikvane позволява на вашето приложение да изпраща SMS и Viber съобщения програмно и да следи статуса на доставка в реално време. То спазва проста REST конвенция с JSON тяло на заявките и стандартни HTTP кодове на отговор.

Свойство	Стойност
Базов URL	https://app.povikvane.com
Протокол	HTTPS (задължителен за всички API заявки)
Content-Type	application/json — задължителен при всички POST заявки
Кодиране	UTF-8
Удостоверяване	API ключ (Bearer токен) + Service ID

i Откъде да вземете данните си за достъп: Вашият API ключ и Service ID се генерират от таблото за управление на акаунта ви в Настройки → API. Пазете API ключа в тайна — третирайте го като парола и никога не го включвайте в код, който се изпълнява в браузъра или от страна на клиента.

API-то предоставя две крайни точки:

Метод	Път	Предназначение
POST	https://app.povikvane.com/public-api/v1/sms	Изпращане на SMS или Viber съобщение
GET	https://app.povikvane.com/public-api/v1/sms/{id}	Проверка на статуса на доставка на съобщение

2 Първи стъпки

Изпратете първото си съобщение за под минута с минималния cURL пример по-долу. Заменете примерните стойности с реалните си данни за достъп:

```
bash

curl -X POST https://app.povikvane.com/public-api/v1/sms \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "service-id": "YOUR_SERVICE_ID",
    "message": {
      "to": "+359888123456",
      "text": "Здравейте! Вашата поръчка беше изпратена.",
      "channel": "sms"
    }
  }'
```

Успешният отговор изглежда така:

```
json

{
  "data": {
    "type": "sms",
    "id": "f47ac10b-58cc-4372-a567-0e02b2c3d479",
    "attributes": {
      "send-at": "2026-06-03 14:00:00",
      "status": "queued_on_smsc",
      "status_detail": "queued",
      "created_at": "2026-06-03 14:00:00",
      "submitted_at": "2026-06-03 14:00:00",
      "delivered_at": null
    },
    "links": {
      "self": "https://app.povikvane.com/public-api/v1/sms/f47ac10b-..."
    }
  }
}
```

- i** Запазете стойността data.id. Тя ще ви е необходима, за да проверявате статуса на доставка (Раздел 8), или може да настроите уебхук, за да получавате актуализации на статуса автоматично (Раздел 10).

3 Удостоверяване

Всяка заявка трябва да съдържа два елемента за удостоверяване:

Елемент	Как се подава	Описание
API ключ	HTTP хедър: Authorization: Bearer	Уникален таен ключ, генериран от таблото ви. Никога не го споделяйте и не го вграждайте в код от страна на клиента.
Service ID	POST: JSON тялото като "service-id" GET: URL параметър ?service-id= или хедър x-service-id:	Уникален идентификатор на вашата услуга, сдвоен с API ключа. Двата трябва да съвпадат; несъответстващи двойки се отхвърлят.

⚠ Внимание: И двата елемента са задължителни при всяка заявка. Липсващ или невалиден API ключ връща HTTP 401 Unauthorized. Липсващ Service ID връща HTTP 400 Bad Request. API ключ, деактивиран от таблото, също връща HTTP 401.

Всеки отговор включва хедър X-Request-Id — уникален идентификатор на конкретната заявка. Записвайте го при вас; посочването му към поддръжката прави проследяването на всеки проблем много по-бързо. По желание може да изпратите собствен хедър X-Request-Id и той ще бъде върнат в отговора.

4 Изпращане на съобщение

POST <https://app.povikvane.com/public-api/v1/sms>

Приема съобщение и го препраща към оператора веднага. Чрез API-то няма планирано или отложено изпращане.

Минимална заявка (SMS)

```
http
POST https://app.povikvane.com/public-api/v1/sms HTTP/1.1
Host: app.povikvane.com
Content-Type: application/json
Authorization: Bearer YOUR_API_KEY

{
  "service-id": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
  "message": {
    "to": "+359888123456",
    "text": "Вашият код за потвърждение е 482910."
  }
}
```

Viber съобщение

```

http

POST https://app.povikvane.com/public-api/v1/sms HTTP/1.1
Host: app.povikvane.com
Content-Type: application/json
Authorization: Bearer YOUR_API_KEY

{
  "service-id": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
  "message": {
    "to": "+359888123456",
    "text": "Здравей, Мария! Часът ти е потвърден за утре в 10:00.",
    "channel": "viber"
  }
}

```

i Viber: Задайте "channel": "viber", за да доставите чрез Viber. Всички специфични за Viber параметри за маршрутизация се конфигурират автоматично от платформата — не е нужно да ги задавате.

5 Параметри на заявката

Параметър	Тип	Задъл.	Описание
service-id	string	Да	Вашият Service ID от таблото. Подава се в JSON тялото.
message.to	string	Да	Телефонен номер на получателя в международен формат, включително кода на държавата. Пример: +359888123456. Нормализира се автоматично. Неподдържани формати — HTTP 400.
message.text	string	Да	Текст на съобщението. Не може да е празен (HTTP 400). Препоръчителен максимум: 1 600 знака.
message.channel	string	Не	Канал за доставка. Допустими стойности: "sms" (по подразбиране) или "viber". Всяка друга стойност се третира като "sms".

i Формат на телефонния номер: Винаги включвайте пълния международен код за избиране. За български номера използвайте +359, последван от 9-цифрения абонатен номер без водеща нула. Водещо 00 се приема и нормализира: 00359888123456 → +359888123456.

6 Идемпотентност (безопасни повторни опити)

Мрежови проблеми могат да ви оставят в неяснота дали една заявка е била обработена. Незадължителният хедър Idempotency-Key предотвратява случайно изпращане на едно и също съобщение два пъти при повторен опит.

Как работи

- Добавете уникален низ като хедър Idempotency-Key (UUID v4 е идеален избор).
- Повторен опит със същия ключ и тяло в рамките на 24 часа връща кеширания отговор — без ново съобщение, без нов кредит. Отговорът носи хедър Idempotent-Replayed: true.
- Същият ключ с различно тяло се отхвърля с HTTP 422.
- Ако повторен опит пристигне докато първата заявка се обработва → HTTP 409 с Retry-After: 1.
- Заявки без хедъра не са засегнати.

Формат на ключа: до 255 знака, само от букви, цифри и знаците _ - : . Ключовете са в обхвата на вашия акаунт. Невалиден ключ → HTTP 400.

Пример с Idempotency-Key

```
http
POST https://app.povikvane.com/public-api/v1/sms HTTP/1.1
Host: app.povikvane.com
Content-Type: application/json
Authorization: Bearer YOUR_API_KEY
Idempotency-Key: 550e8400-e29b-41d4-a716-446655440000

{
  "service-id": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
  "message": {
    "to": "+359888123456",
    "text": "Вашата поръчка #12345 беше изпратена.",
    "channel": "sms"
  }
}
```

- **Добра практика:** Генерирайте UUID преди заявката и го запазете заедно със запис за поръчката. Използвайте същия UUID при всеки повторен опит и нов UUID за всяко ново съобщение.

7 Успешен отговор

Успешното изпращане връща HTTP 200 OK:

```
json
{
  "data": {
    "type": "sms",
    "id": "f47ac10b-58cc-4372-a567-0e02b2c3d479",
    "attributes": {
      "send-at": "2026-06-03 14:00:00",
      "status": "queued_on_smsc",
      "status_detail": "queued",
      "created_at": "2026-06-03 14:00:00",
      "submitted_at": "2026-06-03 14:00:00",
      "delivered_at": null
    },
    "links": {
      "self": "https://app.povikvane.com/public-api/v1/sms/f47ac10b-..."
    }
  }
}
```

Поле	Тип	Описание
data.type	string	Винаги "sms", дори когато каналът е Viber.
data.id	string (UUID)	Уникален идентификатор на съобщението.
data.attributes.send-at	string	Време на изпращане YYYY-MM-DD HH:MM:SS (UTC).
data.attributes.status	string	Винаги queued_on_smsc непосредствено след изпращане.
data.attributes.status_detail	string	По-детайлен статус. При изпращане винаги "queued".
data.attributes.created_at	string	Кога е създаден записът (UTC).
data.attributes.submitted_at	string	Кога платформата е приела съобщението (UTC).
data.attributes.delivered_at	string null	Винаги null в отговора при изпращане.
data.links.self	string (URL)	URL за извличане на текущия статус чрез GET.

8 Проверка на статуса на съобщение

GET /public-api/v1/sms/{id}

Извлича текущия статус на доставка на вече изпратено съобщение. Заменете {id} с UUID-а, върнат при изпращането.

Вариант А — Service ID като параметър в URL (препоръчително)

```
http
```

```
GET https://app.povikvane.com/public-api/v1/sms/f47ac10b-...?service-id=alb2c3d4-... HTTP/1.1
Host: app.povikvane.com
Authorization: Bearer YOUR_API_KEY
```

Вариант Б — Service ID като хедър

```
http
```

```
GET https://app.povikvane.com/public-api/v1/sms/f47ac10b-... HTTP/1.1
Host: app.povikvane.com
Authorization: Bearer YOUR_API_KEY
x-service-id: alb2c3d4-e5f6-7890-abcd-ef1234567890
```

Примерен отговор (доставено съобщение)

```
json
```

```
{
  "data": {
    "type": "sms",
    "id": "f47ac10b-58cc-4372-a567-0e02b2c3d479",
    "attributes": {
      "send-at": "2026-06-03 14:00:00",
      "status": "delivered_to_handset",
      "status_detail": "delivered"
    },
    "links": { "self": "https://app.povikvane.com/..." }
  }
}
```

Препоръка: Уебхуквете (Раздел 10) изпращат крайния статус на доставка към вашата крайна точка в момента, в който той пристигне от оператора. Те са по-ефективни от проверка в цикъл и намаляват излишните API заявки.

9

Статуси и жизнен цикъл на съобщенията

Статуси, връщани от API-то (status)

Статус	Значение
queued_on_smsc	Съобщението е прието и препратено към оператора. Потвърждението за доставка предстои. Винаги е статусът непосредствено след успешно изпращане.

Статус	Значение
delivered_to_handset	Операторът потвърди, че съобщението е доставено до устройството на получателя.
not_delivered_to_handset	Съобщението не беше доставено. Причини: операторът го отхвърли, номерът е невалиден или изключен, Viber не е инсталиран, TTL изтекъл.

Жизнен цикъл на статуса

```

queued_on_smsc → delivered_to_handset (успешен път)
queued_on_smsc → not_delivered_to_handset (път при неуспех)

```

Полето status_detail

status_detail	Когато status е	Значение
queued	queued_on_smsc	Прието и препратено към оператора; очаква се DLR.
delivered	delivered_to_handset	Получена DLR: съобщението е достигнало устройството.
(пропуснато)	not_delivered_to_handset	При недоставяне не се връща status_detail.

Съответствие между имената при уебхук и GET

Резултат	status при GET	data.status при уебхук
Доставено	delivered_to_handset	delivered
Недоставено	not_delivered_to_handset	failed
Все още в движение	queued_on_smsc	(уебхук още не е изпратен)

10 Уебхукове (push известия)

Конфигурирайте URL за уебхук в таблото си на Настройки → API → Webhook URL, за да получавате автоматични HTTP POST известия, когато съобщение достигне краен статус на доставка.

Кога се изпраща уебхук?

Известията се задействат само когато съобщението достигне едно от двата крайни състояния: delivered или failed.

Тяло на уебхука

```

json

{
  "event": "message.status_updated",
  "data": {
    "id": "f47ac10b-58cc-4372-a567-0e02b2c3d479",
    "status": "delivered",
    "channel": "sms",
    "recipient": "+359888123456",
    "timestamp": "2026-06-03T14:01:23.000Z",
    "error": null
  }
}

```

Поле	Тип	Описание
event	string	Винаги "message.status_updated".
data.id	string (UUID)	Идентификатор на съобщението. Съвпада с data.id от отговора при изпращане.
data.status	string	"delivered" или "failed".
data.channel	string	"sms" или "viber".
data.recipient	string	Телефонният номер на получателя в международен формат.
data.timestamp	string (ISO 8601)	Кога е настъпило събитието за статус.
data.error	string null	Четима причина за неуспеха при "failed"; null при успех.

Изисквания към вашата крайна точка

- Трябва да е достъпна по http:// или https:// от публичния интернет (не localhost).
- Трябва да отговори с HTTP 2xx в рамките на 5 секунди. Известието не се повтаря автоматично при неуспех.
- Конфигурира се през таблото — не през самото API.

i Добри практики: Върнете HTTP 200 веднага щом получите заявката, след което обработете данните във фоновата задача. Проверявайте дали вече сте обработили data.id преди да действате — едно и също известие може рядко да пристигне повече от веднъж.

⚠ Важно: Уебхуковете се изпращат еднократно без автоматично повторение. Третирайте крайната точка GET (Раздел 8) като резервен механизъм и периодично сверявайте.

11 Лимити на заявките

API-то налага два независими лимита върху крайната точка POST за изпращане, с плъзгащ прозорец от 60 секунди:

Ниво на лимит	Обхват	Лимит	Прозорец
На фирма	Вашият фирмен акаунт	400 съобщения	60 секунди
За цялата платформа	Всички клиенти заедно	4 000 съобщения	60 секунди

При надхвърляне API-то връща HTTP 429 Too Many Requests с хедър Retry-After.

```
http
HTTP/1.1 429 Too Many Requests
Retry-After: 15
Content-Type: application/json

{
  "errors": [{
    "status": "429",
    "title": "Rate limit exceeded",
    "detail": "You have exceeded the rate limit of 400 messages per minute."
  }]
}
```

Лимит за GET

Поддържайте проверките в рамките на ориентир от 600 заявки на 60 секунди на Service ID. Препоръчваме уебхуковете като основен механизъм за известяване.

Добри практики

- Спазвайте Retry-After — изчакайте поне толкова секунди. Незабавните повторни опити забавят нулирането.
- Използвайте Idempotency-Key при повторни опити (Раздел 6), за да не удържате кредит два пъти.
- Прилагайте експоненциално нарастващо изчакване след повтарящи се неуспехи.
- Разпределяйте масовите изпращания във времето.
- Използвайте уебхукове вместо периодична проверка.

12 Кредитна система

Всяко изпратено съобщение (SMS или Viber) използва 1 кредит от баланса на вашата фирма. Кредитите се проверяват и удържат атомарно в момента на заявката.

Сценарий	Ефект	Бележки
Съобщението е прието и изпратено	-1 кредит	Удържа се веднага при приемане на заявката.
Недостатъчно кредити (HTTP 402)	0 кредита	Заявката се отхвърля преди да бъде удържан кредит.
Операторът изрично отхвърля съобщението (HTTP 502)	+1 кредит (върнат)	Кредитът се връща автоматично при ясен отказ от оператора.
Връзката с оператора изтече (HTTP 504)	-1 кредит (не се връща)	Резултатът е неизвестен; кредитът не се връща автоматично.

⚠ HTTP 504: При HTTP 504 (Gateway Timeout) съобщението може да е изпратено или не. Ако често виждате тази грешка, свържете се с поддръжката.

i Следете баланса си: Оставащите ви кредити се показват в горната лента на таблото и в Настройки → Плащане. Зареждайте предварително, за да избегнете прекъсвания.

13 Справочник на грешките

Всички отговори с грешка споделят тази JSON структура:

```
json
{
  "errors": [
    {
      "status": "400",
      "title": "Bad Request",
      "detail": "Missing message.to or message.text"
    }
  ]
}
```

HTTP код	Заглавие	Кога възниква
400	Bad Request	Липсващи или невалидни параметри: service-id, message.to, message.text, невалиден телефонен номер, неправилен JSON или невалиден Idempotency-Key.
401	Unauthorized	Липсващ, грешен или деактивиран API ключ; или Service ID не съвпада с API ключа.

HTTP код	Заглавие	Кога възниква
402	Payment Required	Недостатъчно кредити. Не е удържан кредит. Заредете баланса си.
404	Not Found	Няма съобщение с подадения ID при GET (връща се и когато съобщението е на друга фирма).
409	Conflict	Заявка с Idempotency-Key все още се обработва. Спазете Retry-After: 1.
422	Unprocessable Entity	Idempotency-Key е използван повторно с различно тяло.
429	Too Many Requests	Надхвърлен лимит. Прочетете Retry-After. Вижте Раздел 11.
500	Internal Server Error	Неочаквана грешка в платформата. Изчакайте и опитайте отново.
502	Bad Gateway	Операторът изрично отхвърли съобщението. Кредитът ви се връща.
504	Gateway Timeout	Връзката с оператора изтече. Резултатът е неизвестен; кредитът не се връща.

14 Правила за валидация

Поле	Правило
service-id	Задължително. Трябва да съвпада с активен API ключ. Чувствително към главни/малки букви.
message.to	Задължително. Валиден телефонен номер (libphonenumber-js). Нормализира се автоматично. Невалидни → HTTP 400.
message.text	Задължително. Не може да е празно (HTTP 400).
message.channel	Незадължително. "sms" (по подразбиране) или "viber". Всяка друга → "sms".
Idempotency-Key	Незадължително. До 255 знака: A-Z a-z 0-9 _ - : . В обхвата на акаунта. Прозорец: 24 часа.

Препоръчителна дължина на текста на съобщението

Канал	Препоръчителен максимум	Бележки
SMS	1 600 знака	Дългите съобщения се разделят на части (160 знака ASCII, 70 за Unicode/кирилица). Всяко изпращане = 1 кредит.
Viber	1 000 знака	Поддържа пълен Unicode и емоджи. 1 кредит.

15 Примери с код**cURL — Изпращане на съобщение**

```
bash
curl -X POST https://app.povikvane.com/public-api/v1/sms \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -H "Idempotency-Key: 550e8400-e29b-41d4-a716-446655440000" \
  -d '{
    "service-id": "YOUR_SERVICE_ID",
    "message": {
      "to": "+359888123456",
      "text": "Здравейте! Вашата поръчка #12345 беше изпратена.",
      "channel": "sms"
    }
  }'
```

cURL — Проверка на статуса

```
bash
curl "https://app.povikvane.com/public-api/v1/sms/MESSAGE_ID\
?service-id=YOUR_SERVICE_ID" \
-H "Authorization: Bearer YOUR_API_KEY"
```

PHP — Изпращане на съобщение

```
php
function sendPovikvaneMessage(
    string $apiKey, string $serviceId,
    string $to, string $text,
    string $channel = 'sms',
    ?string $idempotencyKey = null
): array {
    $url = 'https://app.povikvane.com/public-api/v1/sms';
    $payload = json_encode([
        'service-id' => $serviceId,
        'message' => ['to' => $to, 'text' => $text, 'channel' => $channel],
    ]);
    $headers = [
        'Content-Type: application/json',
        'Authorization: Bearer ' . $apiKey,
    ];
    if ($idempotencyKey !== null)
        $headers[] = 'Idempotency-Key: ' . $idempotencyKey;
    // ... curl_init, curl_exec, json_decode ...
}
```

Node.js — Изпращане на съобщение

```
javascript

async function sendMessage({ apiKey, serviceId, to, text,
  channel = "sms", idempotencyKey }) {
  const headers = {
    "Content-Type": "application/json",
    Authorization: `Bearer ${apiKey}`,
  };
  if (idempotencyKey) headers["Idempotency-Key"] = idempotencyKey;

  const res = await fetch(
    "https://app.povikvane.com/public-api/v1/sms", {
      method: "POST", headers,
      body: JSON.stringify({
        "service-id": serviceId,
        message: { to, text, channel },
      }),
    });
  const data = await res.json();
  if (res.ok) return { ok: true, id: data.data.id };
  return { ok: false, code: res.status };
}
```

Python — Изпращане на съобщение

```
python

import requests

def send_message(api_key, service_id, to, text,
  channel="sms", idempotency_key=None):
  headers = {
    "Content-Type": "application/json",
    "Authorization": f"Bearer {api_key}",
  }
  if idempotency_key:
    headers["Idempotency-Key"] = idempotency_key

  res = requests.post(
    "https://app.povikvane.com/public-api/v1/sms",
    headers=headers,
    json={"service-id": service_id,
      "message": {"to": to, "text": text, "channel": channel}},
    timeout=30,
  )
  data = res.json()
  if res.status_code == 200:
    return {"ok": True, "id": data["data"]["id"]}
  return {"ok": False, "code": res.status_code}
```

16 Поддръжка

Ако имате въпроси относно интеграцията с публичното API на Povikvane, нужда от изясняване на статус на доставка или срещате постоянни грешки 500/504, свържете се с нас на contact@povikvane.com.

i За по-бързо проследяване: Когато съобщавате за проблем, посочете data.id на съобщението и хедъра X-Request-Id от отговора — те ни позволяват да проследим заявката ви бързо.

*Povikvane Public API — Ръководство за интеграция за клиенти · v1.3
Май 2026*