



API Документація

1. Формування токена

2. SMS

2.1 Відправка SMS повідомлення по API

2.1.1 Приклад json запиту по відправці SMS

2.1.2 Блок SMS

2.1.3 Блок Отримувача

2.1.4 Коди відповіді та причини можливих помилок (Responses)

2.2 Звіт по SMS - отримати список надісланих SMS з інформацією

2.3 Перевірка роботи SMS API

2.4 Перевірка SMS API версії

3. VIBER

3.1 Відправка Viber повідомлення по API

3.1.1 Приклад json запиту по відправці Viber повідомлення

3.1.2 Блок viber

3.1.3 Блок recipients

3.2. Viber. Карусель - відправка повідомлень

3.2.1 Карусель. Схема POST /viber запиту json

3.2.2. Блок viber

3.2.3. Блок carousel

3.2.4. Блок recipients

3.3. Viber. Опитувальник - відправка повідомлень

3.3.1. Схема POST /viber запиту json: опитувальник.

3.3.2. Блок Viber

3.3.3. Блок recipients

3.3.4. Блок list

3.4. Viber. Отримати список надісланих повідомлень Viber з інформацією

3.4.1. Приклад json запиту на отримання списку Viber повідомлень

3.4.2 Опис отриманих даних на GET запит списку Viber повідомлень:

3.5 Отримати конкретну інформацію по Viber - повідомленням

3.5.1. Приклад json запиту на отримання списку Viber повідомлень

3.5.2. Перевірка роботи VIBER API

3.5.3. Перевірка VIBER API версії

4. Two Way Chat. Viber

4.1 Створення двостороннього чату Two Way

4.2 Отримати список двосторонніх чатів Two Way

4.3 Отримати конкретну інформацію про двосторонній чат Two Way

4.4 Оновити конкретну інформацію про двосторонній чат Two Way

4.5 Видалити конкретний двосторонній чат із пов'язаними повідомленнями Two Way

4.6 Отримати список двосторонніх повідомлень Two Way

4.7 Отримати інформацію про конкретному ID двостороннього чату Two Way

5. RCS

5.1 Відправка RCS повідомлення



[5.1.1 Приклад json по rcs](#)

[5.1.2 Опис параметрів RCS повідомлення](#)

[5.1.3 Коди відповіді RCS](#)

[5.2 Отримання статусу RCS повідомлення](#)

[6. Каскадні розсилки \(cascade\)](#)

[6.1 Відправка каскадного повідомлення](#)

[6.1.1. Приклад json запиту по відправці каскаду](#)

[6.1.2. Блок channels](#)

[6.1.3 Блок recipients](#)

[6.1.5. Блок sms](#)

[6.1.6 Блок viber](#)

[6.2. Отримати статус по каскадній розсилці](#)

[7. Реєстрація Альфа - імені по запиту](#)

[7.1 Приклад json запиту по відправці запиту на створення AI](#)

[7.2 Коди відповіді при створенні AI](#)

Примітка 1. Розробник може використовувати також опис функціоналу по лінку

<https://client.octotoo.ua/swagger/>

Примітка 2. Спілкування з сервісами здійснюється за допомогою відправки через HTTP API-запитів на задану адресу, використовуючи текстовий формат обміну даними JSON



1.Формування токєну

Для аутентифікації запиту використовується JSON OAuth 2.

При цьому виді аутентифікації необхідно використовувати token, який можна отримати в особистому кабінеті в розділі Профіль. Необхідно включити цей токен у заголовок запиту, наприклад: 'Authorization: Bearer your_token'.

Увага! Обмеження частоти запитів до сервісу має бути не більше ніж 9 запитів на секунду від одного клієнта.

Для початку роботи - сформуйте в особистому кабінет API токен.

1.1 Натисніть на Логін в правому верхньому кутку, у випадаючому списку - виберіть Профіль

1.2 В Полі Токен - скопіюйте повністю дані та передайте вашим розробникам для реалізації API підключення.



2. SMS

2.1 Відправка SMS повідомлення по API

2.1.1 Приклад json запиту по відправці SMS

Необхідно виконати POST запит на url <https://api.octotoo.ua/sms-v3>

Токен для API підключення ви знайдете в особистому кабінеті в Профілі користувача

[1.Формування токenu](#)

Приклад json запиту по відправці SMS

```
{
  "sms": {
    "from": "string",
    "text": "string",
    "nameDelivery": "string",
    "lifetime": 180,
    "sendFromDate": "2025-05-05 10:00:00",
    "sendToDate": "2025-05-05 10:00:00"
  },
  "recipients": [
    {
      "phone": 380961111111
    }
  ]
}
```

Далі більш детально описані вимоги до параметрів.

* зірочкою позначенні поля обов'язкові для заповнення.

2.1.2 Блок SMS

Sms{	Об'єкт, що містить дані для формування SMS-повідомлення.		
Параметр	Формат даних	Обов'язково/ Опціонально	Опис поля
from*	string	Обов'язково	Ім'я відправника - Альфа Ім'я
text*	string	Обов'язково	Текст SMS повідомлення
nameDelivery	string	Опціонально	Текст повідомлення
lifetime	integer	Опціонально	Скільки часу (у секундах) система намагатиметься доставити повідомлення default => 3600 maximum => 144400 minimum => 60

			Для смс значення можуть бути в діапазоні 60-144 400 секунд. Якщо користувач не вказав примусове значення, буде встановлено значення автоматично на 3600 секунд.
sendFromDate	string(\$date-time) Format: Y-m-d H:i:s Приклад: 2025-05-05 10:00:00	Опціонально	Дата та час, з якої необхідно запустити відправку повідомлення (планування розсилки).
sendToDate	string(\$date-time) Format: Y-m-d H:i:s Приклад: 2025-05-05 10:00:00	Опціонально	Дата та час, до якої необхідно відправити повідомлення.

2.1.3 Блок Отримувача

"recipients": [minItems: 1 maxItems: 1000 xml: OrderedMap { "wrapped": true } List of message recipients. "recipients": [{ "phone": 380961111111 }, { "phone": 380976328628 }] 		Об'єкт, що містить дані номерів телефонів	
Параметр	Формат даних	Обов'язково/ Опціонально	Опис поля
phone*	integer	Обов'язково	Номер отримувача Приклад: 380961111111 - мінімум 1, максимум 1000



2.1.4 Коди відповіді та причини можливих помилок (Responses)

Код відповіді	Опис	Приклад відповіді json
201 Successful request	Успішна відправка повідомлення	<pre>{ "status": "success", "info": { "380961111111": 4123112 } }</pre>
Код відповіді	Опис	Приклад відповіді json
422 Validation - коди відповіді помилок валідації даних		
code 2 Text cannot be blank	Текст не може бути порожнім	<pre>{ "status": "failed", "errors": [{ "field": "text", "code": 22, "description": "Text cannot be blank." }] }</pre>
code 3, "field": "phone" Phone should contain at most 15 characters	Номер телефону має містити не більше 15 символів	<pre>{ "status": "failed", "invalidRecipients": [{ "recipient": { "phone": 11111111111111100 }, "errors": [{ "field": "phone", "code": 3, "description": "Phone should contain at most 15 characters." }] }], "errors": [{ "field": "recipients", "code": 2, "description": "Must be at least 1 correct recipient." }] }</pre>
code 2 Must be at least 1 correct recipient	Повинен бути принаймні 1 правильний одержувач	<pre>{ "status": "failed", "invalidRecipients": [{ "recipient": { "phone": 11111111111111100 }, "errors": [{ "field": "phone", "code": 3, "description": "Phone should contain at most 15 characters." }] }], "errors": [{ "field": "recipients", "code": 2, "description": "Must be at least 1 correct recipient." }] }</pre>



"code": 31, "field": "phone"	Текстове поле обов'язкове для заповнення.	<pre>{ "status": "failed", "errors": [{ "field": "from", "code": 31, "description": "Originator not found." }] }</pre>
code 21 "field": "from", meaning some trouble with sender name.	Необхідно перевірити, чи коректно вказане та активне Альфа-Ім'я для відправки повідомлень	<pre>{ "status": "failed", "errors": [{ "field": "from", "code": 21, "description": "From should contain at most 32 characters." }] }</pre>
500 Internal server error	Внутрішня помилка сервера	

2.2 Звіт по SMS - отримати список надісланих SMS з інформацією

Необхідно виконати **GET** запит на url <https://api.octotoo.ua/sms-v3>

Токен для API підключення ви знайдете в особистому кабінеті в Профілі користувача

1.Формування токenu

- Якщо необхідно отримати інформацію по списку смс повідомлень (без деталізації по id)
<https://api.octotoo.ua/sms-v3>
- Якщо необхідно отримати інформацію по одній смс, приклад URL
[https://api.octotoo.ua/sms-v3?filter\[id\]\[in\]\[\]=id SMS](https://api.octotoo.ua/sms-v3?filter[id][in][]=id SMS)
- Якщо необхідно отримати інформацію по списку смс повідомлень, вказувати потрібно
відповідно через знак поєднання перелік id смс повідомлень, приклад URI

[https://api.smsclub.mobi/sms-v3?filter\[id\]\[in\]\[\]=id SMS&filter\[id\]\[in\]\[\]=id SMS](https://api.smsclub.mobi/sms-v3?filter[id][in][]=id SMS&filter[id][in][]=id SMS)

ВІДПОВІДЬ. Приклад успішної відповіді на **GET** запит з заповненими даними



```
{
  "status": "success",
  "data": [
    {
      "id": 111111,
      "state": 0,
      "send_date": "2024-02-27 13:38:19",
      "phone": 380961111111,
      "price": 0.5,
      "MCC": 255,
      "MNC": 6,
      "count_sms": 1,
      "currency": "UAH"
    },
    {
      "id": 111112,
      "state": 1,
      "send_date": "2024-02-27 13:35:19",
      "phone": 380961111112,
      "price": 0.5,
      "MCC": 255,
      "MNC": 6,
      "count_sms": 1,
      "currency": "UAH"
    },
    {
      "id": 111113,
      "state": 3,
      "send_date": "2024-02-27 13:34:19",
      "phone": 380961111113,
      "price": 0.5,
      "MCC": 255,
      "MNC": 6,
      "count_sms": 1,
      "currency": "UAH"
    }
  ]
}
```

Приклад успішної відповіді на **GET** запит з пустими даними

```
{
  "status": "success",
  "data": []
}
```

У відповідь (Responses), можливо отримати наступні дані про SMS:

Отримані дані (з прикладу)	Формат даних	Опис
----------------------------	--------------	------



{ "status": "success",	Статус запиту, в даному випадку "успішний"	
"data": [	Далі відображаються деталі по кожному повідомленню	
"id": 111112,	integer	ID повідомлення в системі
"state": 1,	integer	0 - (message sent) повідомлення надіслано; 1 - (message delivered) повідомлення доставлено; 2 - (message undeliverable) повідомлення не доставлено; 3 - (expired, lifetime exceeded) термін дії минув, не вдалось в результаті доставити повідомлення.
"send_date": "2024-02-27 13:35:19",	string(\$date-tim)	Дата відправки повідомлення
"phone": 38096111112,	string	Номер телефону, на який відправлене повідомлення
"price": 0.5,	number(\$float)	Ціна за повідомлення
"MCC": 255, "MNC": 6,	number(\$integer)	Напрямок країни та оператора, на який було відправлене повідомлення
"count_sms": 1,	number(\$integer)	Кількість частин в повідомленні
"currency": "UAH",	string	Валюта, в якій сплачується плата за повідомлення (price)

2.3 Перевірка роботи SMS API

Необхідно виконати GET запит на url <https://api.octotoo.ua/sms-v3/health>

У відповідь (Responses), можливо отримати наступні статуси SMS:

200 API works in default mode - сервіс працює;

503 Service Temporary Unavailable - сервіс тимчасово недоступний.

2.4 Перевірка SMS API версії

Необхідно виконати GET запит на url <https://api.octotoo.ua/sms-v3/version>

У відповідь (Responses), можливо отримати наступну інформацію: Code 200

```
{
  "status": "success",
  "version": "1.0.0"
}
```



3. VIBER

3.1 Відправка Viber повідомлення по API

3.1.1 Приклад json запиту по відправці Viber повідомлення

Необхідно виконати **POST** запит на url <https://api.octotoo.ua/viber>

Токен для API підключення ви знайдете в особистому кабінеті в Профілі користувача

[1.Формування токenu](#)

Приклад json запиту по відправці Viber повідомлення

```
{
  "viber": {
    "from": "string",
    "text": "string",
    "device": "all",
    "lifetime": 86000,
    "nameDelivery": "string",
    "sendFromDate": "2025-05-05 10:00:00",
    "sendToDate": "2025-05-05 10:00:00",
    "button": {
      "text": "string",
      "url": "string"
    },
  },
  "image": {
    "url": "string"
  },
  "video": {
    "url": "string",
    "callback": "string",
    "preview": "string",
    "buttonCaption": "string",
    "fileSize": 0,
    "duration": 0
  },
  "file": {
    "url": "string",
    "name": "string",
    "ext": "doc"
  },
  "twoWay": {
    "chatName": "string"
  },
  "recipients": [
    {
      "phone": 380961111111
    }
  ]
}
```

```
}
}
```

Далі більш детально описані вимоги до параметрів.

* зірочкою позначенні поля обов'язкові для заповнення.

3.1.2 Блок viber

"viber": {		Об'єкт, що містить дані для формування Viber-повідомлення.	
Параметр	Формат даних	Обов'язково/ Опціонально	Опис поля
from*	string	Обов'язково	Ім'я відправника - Альфа Ім'я
text*	string	Обов'язково	Текст повідомлення
device	[all, main]	Опціонально	Вказівник на пристрій, на який доставляється повідомлення. На всі присторої користувача, на яких встановлений viber (all), чи тільки на основний пристрій (main).
lifetime	default: 86000 minimum: 60 maximum: 86000	Опціонально	Час, протягом якого повідомлення система намагатиметься доставити.
nameDelivery	string	Опціонально	Довільна назва розсилки.
sendFromDate	string(\$date-time) Format: Y-m-d H:i:s Приклад: 2025-05-05 10:00:00	Опціонально	Дата та час, з якої необхідно запустити відправку повідомлення (планування розсилки).
sendToDate	string(\$date-time) Format: Y-m-d H:i:s Приклад: 2025-05-05 10:00:00	Опціонально	Дата та час, до якої необхідно відправити повідомлення.
"button": {		Кнопка в Viber повідомленні	
text	string	Опціонально	Максимальна довжина: 50.
url	string	Опціонально	Посилання, за яким переходить користувач після натискання кнопки.

"image": {		Зображення в Viber повідомленні. Опціональне. Але якщо його вказати в запит, то саме url являється обов'язковим.	
url*	string(\$uri)	Обов'язково	Посилання, яке вказує на зображення.
"video": {		Відео в Viber повідомленні. Опціональне	
url*	string(\$uri)	Обов'язково	Посилання, яке вказує на відео.
callback	string(\$uri)	Опціонально	URL-адреса, яка відкривається при натисканні кнопки.
preview*	string	Обов'язково	URL-адреса зображення, яке буде використано для попереднього перегляду відео.
buttonCaption	string	Опціонально	Текст кнопки під відео.
fileSize*	integer	Обов'язково	Розмір відеофайлу в мегабайтах (Мб).
duration	integer	Опціонально	Тривалість відео в секундах.
"file": {		Файл у повідомленні. Опціональне.	
url*	string(\$uri)	Обов'язково	Посилання, яке вказує на файл.
name*	string	Обов'язково	Ім'я файлу.
ext*	doc	Обов'язково	Розширення файлу. Приклад: [doc, docx, rtf, dot, dotx, odt, odf, fodt, txt, info, pdf, xps, pdax, eps, xls, xlsx, ods, fods, csv, xlsx, xltx]
"twoWay": {		Можливість завантаження двосторонніх чатів для спілкування з клієнтами. Блок більш детально описаний	
chatName	string maxLength: 50	Опціонально	Назва чату.

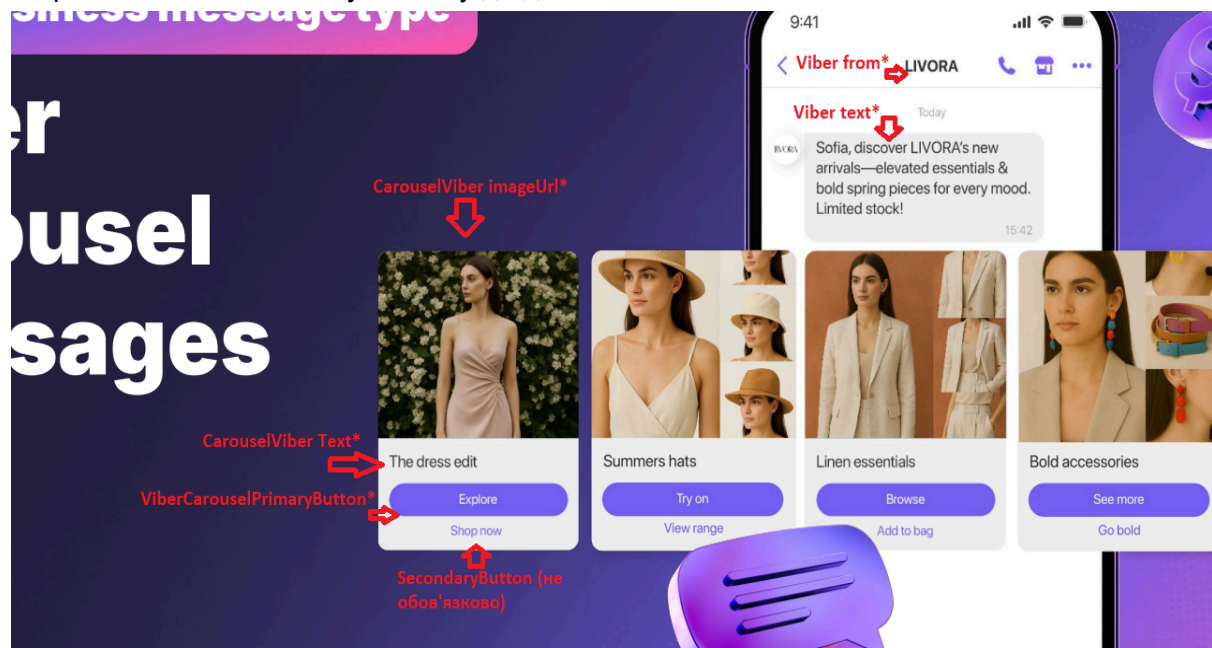
3.1.3 Блок recipients

"recipients": [minItems: 1 maxItems: 1000	Об'єкт, що містить дані номерів телефонів - мінімум 1, максимум 1000.
---	--

xml: OrderedMap { "wrapped": true } List of message recipients.			
Параметр	Формат даних	Обов'язково/ Опціонально	Опис поля
phone*	integer	Обов'язково	Номер отримувача Приклад: 380961111111.

3.2. Viber. Карусель - відправка повідомлень

Карусель дозволяє користувачеві прокручувати список елементів, кожен з яких містить зображення, опис і кнопку заклику до дії.



Передбачено від 2 до 5 екранів каруселі

Необхідна мінімальна версія додатку Viber у користувача 24.9.0.

Зверніть увагу, що в запиті на відправку по каруселі можуть бути тільки основний блок по viber, carousel, recipients.

3.2.1 Карусель. Схема POST /viber запиту json

Токен для API підключення ви знайдете в особистому кабінеті в Профіль користувача

1.Формування токєну

```
{
  "viber": {
    "from": "string",
    "text": "string",
  }
}
```



```
"device": "all",
"lifetime": 86000,
"nameDelivery": "string",
"sendFromDate": "2025-05-05 10:00:00",
"sendToDate": "2025-05-05 10:00:00",
},
"carousel": {
  "sections": [
    {
      "text": "string",
      "imageUrl": "string",
      "primaryButton": {
        "text": "string",
        "url": "string"
      },
      "secondaryButton": {
        "text": "string",
        "url": "string"
      }
    }
  ]
},
"recipients": [
  {
    "phone": 380961111111
  }
]
}
```

Приклад запиту json для відправки повідомлення формату Карусель у Viber

```
{
  "viber": {
    "from": "testname",
    "text": "Літні знижки на сукні! Розпродаж",
    "carousel": {
      "sections": [
        {
          "text": "Літня сукня",
          "imageUrl": "https://picsum.photos/200/300.jpg",
          "primaryButton": {
            "text": "Купити",
            "url": "https://example.com/dress-sale"
          },
          "secondaryButton": {
            "text": "Перейти",
            "url": "https://example.com"
          }
        }
      ]
    }
  }
}
```

```
        "text": "Літня сукня2",
        "imageUrl": "https://picsum.photos/200/300.jpg",
        "primaryButton": {
            "text": "Купити2",
            "url": "https://smsclub.mobi/"
        },
        "secondaryButton": {
            "text": "Перейти",
            "url": "https://example.com"
        }
    },
    {
        "text": "Літня сукня3",
        "imageUrl": "https://picsum.photos/200/300.jpg",
        "primaryButton": {
            "text": "Купити2",
            "url": "https://smsclub.mobi/"
        },
        "secondaryButton": {
            "text": "Перейти",
            "url": "https://example.com"
        }
    },
    {
        "text": "Літня сукня4",
        "imageUrl": "https://picsum.photos/200/300.jpg",
        "primaryButton": {
            "text": "Купити",
            "url": "https://smsclub.mobi/"
        },
        "secondaryButton": {
            "text": "Перейти",
            "url": "https://example.com"
        }
    },
    {
        "text": "Літня сукня 5",
        "imageUrl": "https://picsum.photos/200/300.jpg",
        "primaryButton": {
            "text": "Купити2",
            "url": "https://smsclub.mobi/"
        },
        "secondaryButton": {
            "text": "Перейти",
            "url": "https://example.com"
        }
    }
]
},
"recipients": [
    {
        "phone": 380971231567
    }
]
```



```

    ]
  }

```

Далі більш детально описані вимоги до параметрів блоку **viber**, **carousel**, **recipients**

* зірочкою позначенні поля обов'язкові для заповнення.

3.2.2. Блок viber

описаний у розділі [3.1.2 Блок viber по лінку](#)

3.2.3. Блок carousel

carousel": { "sections": [{		Об'єкт, що містить дані для формування Каруселі від 2х до 5ти картинок з кнопками (sections)	
Параметр	Формат даних	Обов'язково/ Опціонально	Опис поля
CarouselViber* - головний текст під картинкою			
text*	string	Обов'язково	Текст повідомлення Довжина тексту (символів): Мінімальна: 2 Максимальна: 38
imageUrl*	Valid link to the image.	Обов'язково	Посилання для картинки
ViberCarouselPrimaryButton* - основна кнопка (перша)			
text*	string	Обов'язково	Текст основної кнопки Довжина тексту (символів): Мінімальна: 1 Максимальна: 10
url*	string(\$uri)	Обов'язково	Посилання, за яким переходить користувач після натискання кнопки.
SecondaryButton (опціонально) - наступна кнопка (після першої)			
text*	string	Обов'язково, якщо відправляється цей блок SecondaryButton	Текст основної кнопки Довжина тексту (символів): Мінімальна: 1 Максимальна: 12



url*	string(\$uri)	Обов'язково, якщо відправляється цей блок SecondaryButton	Посилання, за яким переходить користувач після натискання кнопки.
------	---------------	--	---

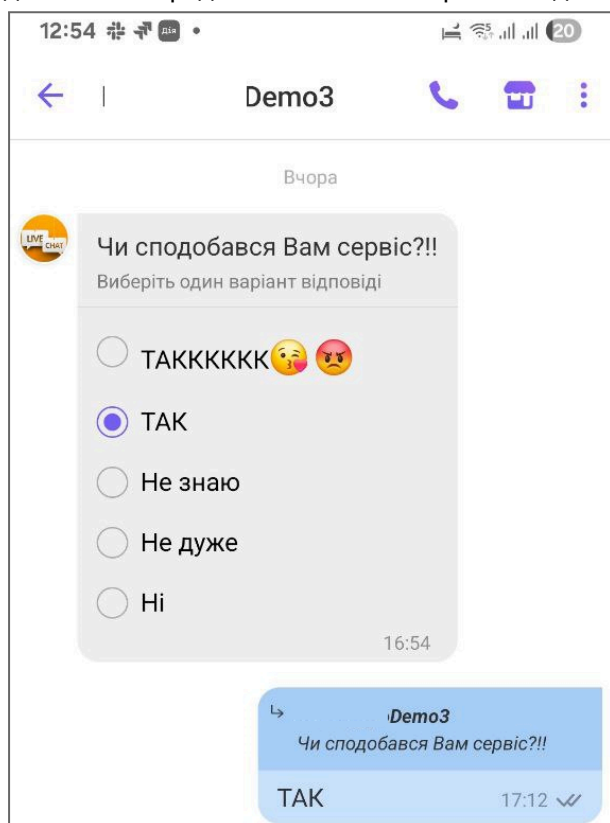
3.2.4. Блок recipients

описаний у розділі [3.1.3 Блок recipients по лінку](#)

Коди відповіді та причини можливих помилок описані у розділі [2.1.4 Коди відповіді та причини можливих помилок \(Responses\)](#)

3.3. Viber. Опитувальник - відправка повідомлень

Опитувальник у Viber — це інтерактивний формат бізнес-повідомлень, розроблений для підвищення ефективності, швидкості та залученості до спілкування з вашими клієнтами. Цей новий тип повідомлень оптимізує процеси прийняття рішень, дозволяючи вам представити до п'яти попередньо визначених варіантів відповіді в одному повідомленні.



Зверніть увагу, що в запиті на відправку Опитувальника можуть бути тільки основний блок по viber, list, recipients.

3.3.1. Схема POST /viber запиту json: опитувальник.

Токен для API підключення ви знайдете в особистому кабінеті в Профілі користувача [1.Формування токenu](#)



```
{
  "viber": {
    "from": "string",
    "text": "string",
    "device": "all",
    "lifetime": 86000,
    "nameDelivery": "string",
    "sendFromDate": "2025-05-05 10:00:00",
    "sendToDate": "2025-05-05 10:00:00",
    "list": {
      "options": [
        "string",
        "string"
      ]
    }
  },
  "recipients": [
    {
      "phone": 380961111111
    }
  ]
}
```

Приклад запиту json для відправки повідомлення формату Опитувальник у Viber

```
{
  "recipients": [
    {
      "phone": "380971234567"
    }
  ],
  "viber": {
    "text": "Чи сподобався Вам сервіс?!!",
    "from": "tesname",
    "list": {
      "options": [
        "Можливо ",
        "ТАК",
        "Не знаю",
        "Не дуже",
        "Ні"
      ]
    }
  }
}
```

3.3.2. Блок Viber

описаний у розділі [3.1.2 Блок viber по лінку](#)



3.3.3. Блок recipients

описаний у розділі [3.1.3 Блок recipients по лінку](#)

3.3.4. Блок list

<code>"list": { "options": [</code>		Об'єкт, що містить дані для формування повідомлення опитувальника	
Параметр	Формат даних	Обов'язково/ Опціонально	Опис поля
options*	string	Обов'язково	Текст варіанту відповіді Довжина тексту (символів): Мінімальна: 2 Максимальна: 50 Кількість відповідей Мінімальна: 2 Максимальна: 5

Після того, як отримувач виконав вибір відповіді у Viber - текст відповіді надійде відправнику в ресурс 2way для подальшої комунікації.

Коди відповіді та причини можливих помилок описані у розділі [2.1.4 Коди відповіді та причини можливих помилок \(Responses\)](#)

3.4. Viber. Отримати список надісланих повідомлень Viber з інформацією

3.4.1. Приклад json запиту на отримання списку Viber повідомлень

Необхідно виконати **GET** запит на url <https://api.octotoo.ua/viber>

Токен для API підключення ви знайдете в особистому кабінеті в Профілі користувача

[1.Формування токenu](#)

ВІДПОВІДЬ. Приклад успішної відповіді на **GET** запит з заповненими даними

```
{
  "status": "success",
  "data": [
    {
      "id": 111111,
      "state": 0,
      "send_date": "2024-02-27 13:38:19",
      "phone": 380961111111,
      "price": 0.5,
      "MCC": 255,
      "MNC": 6,
      "count_sms": 1,
      "currency": "UAH",
      "additionalState": null
    },
    {
      "id": 111112,
      "state": 1,
      "send_date": "2024-02-27 13:35:19",
      "phone": 380961111112,
      "price": 0.5,
      "MCC": 255,
      "MNC": 6,
      "count_sms": 1,
      "currency": "UAH",
      "additionalState": null
    },
    {
      "id": 111113,
      "state": 3,
      "send_date": "2024-02-27 13:34:19",
      "phone": 380961111113,
      "price": 0.5,
      "MCC": 255,
      "MNC": 6,
      "count_sms": 1,
      "currency": "UAH",
      "additionalState": null
    }
  ]
}
```

Приклад успішної відповіді на **GET** запит з пустими даними

```
{
  "status": "success",
  "data": []
}
```



3.4.2 Опис отриманих даних на GET запит списку Viber повідомлень:

Отримані дані (на основі прикладу)	Формат даних	Опис
{ "status": "success",	Статус запиту, в даному випадку "успішний"	
"data": ["id": 111112,	Далі відображаються деталі по кожному повідомленню	
"state": 1,	integer	ID повідомлення в системі
"send_date": "2024-02-27 13:35:19",	integer	Статус повідомлення "state": 0 - відправлено "state": 1, - доставлено "state": 2, - не доставлено "state": 3, - термін доставки пройшов
"phone": 38096111112,	string(\$date-time)	Дата відправки повідомлення
"price": 0.5,	string	Номер телефону, на який відправлене повідомлення
"MCC": 255, "MNC": 6,	number(\$float)	Ціна за повідомлення
"count_sms": 1,	number(\$integer)	Напрямок країни та опер атора, на який було відправлене повідомлення
"currency": "UAH",	string	Кількість повідомлень
"additionalState": null	string	Валюта, в якій сплачується плата за повідомлення (price)
		Додатковий статус

3.5 Отримати конкретну інформацію по Viber - повідомленням

3.5.1. Приклад json запиту на отримання списку Viber повідомлень

Необхідно виконати **GET** запит на url <https://api.octotoo.ua/viber/{viberId}>



Токен для API підключення ви знайдете в особистому кабінеті в Профілі користувача
[1.Формування токenu](#)

ВІДПОВІДЬ. Приклад успішної відповіді на **GET** запит з заповненими даними

```
{
  "status": "success",
  "data": [
    {
      "id": 160812230,
      "state": 0,
      "send_date": "2024-02-27T12:56:17.000Z",
      "phone": 380961111111,
      "price": "1.00000",
      "MCC": 255,
      "MNC": 6,
      "count_sms": 1,
      "currency": "UAH",
      "additionalState": null
    }
  ]
}
```

Опис отриманих даних на **GET** запит по конкретному ID Viber повідомлення [3.4.2 Опис отриманих даних на GET запит списку Viber повідомлень](#)

3.5.2. Перевірка роботи VIBER API

Необхідно виконати **GET** запит на url <https://api.octotoo.ua/viber/health>

Токен для API підключення ви знайдете в особистому кабінеті в Профілі користувача
[1.Формування токenu](#)

У відповідь (Responses), можливо отримати наступні статуси:

200 API works in default mode - сервіс працює;

503 Service Temporary Unavailable - сервіс тимчасово недоступний.

3.5.3. Перевірка VIBER API версії

Необхідно виконати **GET** запит на url <https://api.octotoo.ua/viber/version>

У відповідь (Responses), можливо отримати наступну інформацію:

Code 200 Information about API version

```
{
  "status": "success",
  "version": "1.0.0"
}
```





4. Two Way Chat. Viber

4.1 Створення двостороннього чату Two Way

TwoWayChat{		Створення двостороннього чату	
Параметр	Формат даних	Обов'язково/ Опціонально	Опис поля
description	string maxLength: 50	Обов'язково	Назва двостороннього чату
recipient	string minLength: 8 maxLength: 15	Обов'язково	Адреса телефону одержувача чату.
source_addr	string	Обов'язково	Альфа Ім'я - відправника у Viber.

Приклад запиту

```
{
  "name": "string",
  "recipient": "stringst",
  "source_addr": "string"
}
```

4.2 Отримати список двосторонніх чатів Two Way

Необхідно виконати **GET** запит на url <https://api.octotoo.ua/two-way-chats>

У відповідь (Responses), можливо отримати наступну інформацію:

Code 200 Інформація про успішне надсилання Two Way повідомлення

Приклад відповіді:

```
{
  "status": "success",
  "data": [
    {
      "id": 21,
      "name": "New Test",
      "recipient": 380961111111,
      "source_addr": "Test",
      "user_id": 999,
      "active": 1,
      "last_activity": "2024-06-18 10:20:57",
      "create_date": "2024-06-17 15:14:51"
    },
  ],
}
```




```
{
  "id": 20,
  "name": "Awesome Chat",
  "recipient": 38096222222,
  "source_addr": "Test",
  "user_id": 999,
  "active": 1,
  "last_activity": "2024-06-17 15:07:36",
  "create_date": "2024-06-17 15:07:36"
},
{
  "id": 19,
  "name": "One More Chat",
  "recipient": 38096333333,
  "source_addr": "Test 2",
  "user_id": 999,
  "active": 1,
  "last_activity": "2024-06-17 15:02:27",
  "create_date": "2024-06-17 15:02:27"
}
]
```

Опис отриманих даних на **GET** запит списку Two Way чатів:

Отримані дані	Формат даних	Опис
{ "status": "success", "data": [{ "id": 20, "name": "Awesome Chat", "recipient": 38096222222, "source_addr": "Test", "user_id": 999, "active": 1, "last_activity": "2024-06-17 15:07:36", "create_date": "2024-06-17 15:07:36" }, { "id": 19, "name": "One More Chat", "recipient": 38096333333, "source_addr": "Test 2", "user_id": 999, "active": 1, "last_activity": "2024-06-17 15:02:27", "create_date": "2024-06-17 15:02:27" }] }	string	Успішно відправлені повідомлення
{ "id": 20,	integer	ID повідомлення в системі
"name": "Awesome Chat",	string	Назва чату
"recipient": 38096222222,	integer	Номер телефону, на який відправлене повідомлення
"source_addr": "Test",	string	Альфа Ім'я - відправника у Viber.
"user_id": 999,	integer	ID користувача, з яким ведеться переписка в чаті
"active": 1,	integer	Активний чат/неактивний чат
"last_activity": "2024-06-17 15:07:36",	string(\$date-time)	Остання активність в чаті



"create_date": "2024-06-17 15:07:36" },	string(\$date-time)	Дата створення чату
---	---------------------	---------------------

4.3 Отримати конкретну інформацію про двосторонній чат Two Way

Необхідно виконати **GET** запит на url <https://api.octotoo.ua/two-way-chats/{chatId}>

Варіанти відповіді (Responses)

Код (status)	Опис	Приклад
200 success	Повідомлення знайдено	<pre>{ "status": "success", "data": [{ "id": 21, "name": "Test Chat", "recipient": 380961111111, "source_addr": "Test", "user_id": 999, "active": 1, "last_activity": "2024-06-17 15:14:51", "create_date": "2024-06-17 15:14:51" }] }</pre>
404	Повідомлення не знайдено	<pre>{ "name": "Not Found", "message": "Object not found: {id}", "code": 0, "status": 404 }</pre>

4.4 Оновити конкретну інформацію про двосторонній чат Two Way

Необхідно виконати **PATCH** запит на url <https://api.octotoo.ua/two-way-chats/{chatId}>

Опис	Формат даних	Приклад
------	--------------	---------



Оновити назву двостороннього чату (нова назва чату).	string maxLength: 50	<pre>{ "name": "string" }</pre>
--	-------------------------	---

Варіанти відповіді (Responses)

Код (status)	Опис	Приклад
200 success	Повідомлення знайдено	<pre>{ "status": "success", "data": [{ "id": 21, "name": "Test Chat", "recipient": 380961111111, "source_addr": "Test", "user_id": 999, "active": 1, "last_activity": "2024-06-17 15:14:51", "create_date": "2024-06-17 15:14:51" }] }</pre>
404	Повідомлення не знайдено	<pre>{ "name": "Not Found", "message": "Object not found: {id}", "code": 0, "status": 404 }</pre>
422	Помилка перевірки	<pre>"status": "failed", "errors": [{ "field": "name", "code": 504, "description": "Name must not be equal to \"{Old One}\"." }]</pre>



31	Означає деякі проблеми з ім'ям в Two Way Chat.	
----	--	--

4.5 Видалити конкретний двосторонній чат із пов'язаними повідомленнями Two Way

Необхідно виконати **DELETE** запит на url <https://api.octotoo.ua/two-way-chats/{chatId}>

Варіанти відповіді (Responses)

Код (status)	Опис	Приклад
204	Чат та пов'язані з ним повідомлення успішно видалено.	
404	Повідомлення не знайдено	<pre>{ "name": "Not Found", "message": "Object not found: {id}", "code": 0, "status": 404 }</pre>

4.6 Отримати список двосторонніх повідомлень Two Way

Необхідно виконати **GET** запит на url <https://api.octotoo.ua/two-way-messages>

Варіанти відповіді (Responses)

Код (status)	Опис	Приклад
200 success	Успішна відповідь на запит	
	Приклад відповіді з пустими даними	<pre>{ "status": "success", "data": [] }</pre>

	Приклад відповіді з даними про чат	<pre>{ "status": "success", "data": [{ "id": 16788, "chat_id": 3, "text": "Hello. You have a doctor's appointment today at 12 a.m.", "date": "2024-06-18 16:35:13", "type": "outcoming", "attachment": { "name": "NW93r17171932861717079027322588.png", "type": "image", "url": "https://example.com:9111/smsbucket/NW93r17171932 861717079027322588.png?X-Amz-Content-Sha256=UNSIG NED-PAYLOAD&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Am z-Credential=Wf3qNtOrW4ejjyyT2nIc%2F20240618%2Feu -central-2%2Fs3%2Faws4_request&X-Amz-Date=2024061 8T145608Z&X-Amz-SignedHeaders=host&X-Amz-Expires= 604800&X-Amz-Signature=1221958154a3ce1269a1ec376f 407dbd9e8895afe3817254af67a09a34cbb7c7" } }, { "id": 16857, "chat_id": 3, "text": "Thank you for notification! I'll write it down for myself.", "date": "2024-06-18 16:42:54", "type": "incoming", "attachment": null }, { "id": 17002, "chat_id": 3, "text": "We will be waiting for you. Have a nice day!", "date": "2024-06-18 16:50:01", "type": "outcoming", "attachment": null }] }</pre>
--	---	---

4.7 Отримати інформацію про конкретному ID двостороннього чату Two Way

Необхідно виконати **GET** запит на url <https://api.octotoo.ua/two-way-messages/{messageId}>



Варіанти відповіді (Responses)

Код (status)	Опис	Приклад
200 success	Успішна відповідь на запит	<pre>{ "status": "success", "data": [{ "id": 16857, "chat_id": 3, "text": "Thank you for notification! I'll write it down for myself.", "date": "2024-06-18 16:42:54", "type": "incoming", "attachment": null }] }</pre>
404	Чат не знайдено	<pre>{ "name": "Not Found", "message": "Object not found: {id}", "code": 0, "status": 404 }</pre>
	Приклад відповіді з даними про чат	<pre>{ "status": "success", "data": [{ "id": 16857, "chat_id": 3, "text": "Thank you for notification! I'll write it down for myself.", "date": "2024-06-18 16:42:54", "type": "incoming", "attachment": null }] }</pre>



5. RCS

RCS (Rich Communication Services) — це наступний крок в еволюції SMS, обмін повідомленнями з розширеними функціями.

RCS-повідомлення підтримують передачу файлів і мультимедійного контенту, надсилання голосових повідомлень, дозволяють створювати групові чати та інше.

На відміну від SMS, RCS дозволяє надсилати не тільки текст, але й фотографії.

*Поки що працює лише на Android

5.1 Відправка RCS повідомлення

Необхідно виконати **POST** запит на url <https://api.octotoo.ua/rcs>

Токен для API підключення ви знайдете в особистому кабінеті в Профіль користувача (інструкція по [лінку](#))

5.1.1 Приклад json по rcs

```
{
  "rcs": {
    "from": "Sender",
    "text": "Message text",
    "buttonText": "Button text",
    "buttonUrl": "https://example.com",
    "imgUrl": "https://example.com/image.jpg",
    "sendFromDate": "2024-04-01 10:00:00",
    "sendToDate": "2024-04-01 11:00:00"
  },
  "recipients": [
    {
      "phone": 380961111111
    },
    {
      "phone": 380961111112
    }
  ]
}
```

5.1.2 Опис параметрів RCS повідомлення

Параметр	Формат даних	Обов'язково/ Опціонально	Опис поля
Rcs{ description:			Об'єкт, що містить дані для формування RCS-повідомлення.
from	string	Обов'язково	Ім'я відправника - Альфа Ім'я
text	string	Обов'язково	Текст повідомлення



buttonText	string	Опціонально	Текст кнопки, що надсилається абоненту в повідомленні RCS
buttonUrl	string(\$uri)	Опціонально	URL-адреса кнопки для переходу
imgUrl	string(\$uri)	Опціонально	URL-адреса медіафайлу (зображення), надісланого клієнту
rcsTtl	integer	Опціонально	Скільки часу (у секундах) слід намагатися доставити повідомлення Мінімум 60 секунд, максимальне значення 144400 'defaultRcs' => 3600, 'maxRcs' => 144400, 'minRcs' => 60,
sendFromDate	DateDatestring(\$date-time) Format: Y-m-d H:i:s Приклад: 2025-05-05 10:00:00	Опціонально	Дата та час, з якої необхідно запустити відправку повідомлення (планування розсилки).
sendToDate	DateDatestring(\$date-time) Format: Y-m-d H:i:s Приклад: 2025-05-05 10:00:00	Опціонально	Дата та час, до якої необхідно відправити повідомлення.
nameDelivery	string	Опціонально	Назва розсилки
Recipients[]		Обов'язково	Список одержувачів RCS-повідомлень.
phone integer	phone integer Приклад: 380961111111	Обов'язково	Номер одержувача.

5.1.3 Коди відповіді RCS

Код відповіді	Опис	Приклад відповіді json
201 Successful request	Успішна відправка повідомлення	<pre>{ "status": "success", "info": { "380961111111": 4123112 } }</pre>

Код відповіді	Опис	Приклад відповіді json
422 Validation error - коди відповіді помилок валідації даних		
code: 10 Text field is required.	Текстове поле обов'язкове для заповнення. Не може бути пустим	<pre>{ "status": "failed", "errors": [{ "field": "text", "code": 10, "description": "Text field is required." }] }</pre>
code: 3 Phone value must contain a maximum of 15 characters	Значення телефону має містити максимум 15 символів	<pre>{ "status": "failed", "invalidRecipients": [{ "recipient": { "phone": 11111111111111100 }, "errors": [{ "field": "phone", "code": 3, "description": "Phone value must contain a maximum of 15 characters." }] }] }</pre>
500 Internal server error	Внутрішня помилка сервера	

5.2 Отримання статусу RCS повідомлення

Необхідно виконати **GET** запит на url <https://api.octotoo.ua/rcs/{rcsId}>

Токен для API підключення ви знайдете в особистому кабінеті в Профілі користувача (інструкція по [лінку](#))

- Якщо необхідно отримати інформацію по одному RCS повідомленню, приклад URL [https://api.smsclub.mobi/rcs?filter\[id\]\[in\]\[\]=id](https://api.smsclub.mobi/rcs?filter[id][in][]=id) RCS



- Якщо необхідно отримати інформацію по списку sms повідомлень, вказувати потрібно відповідно через знак поєднання перелік id sms повідомлень, приклад URI [https://api.smsclub.mobi/rcs?filter\[id\]\[in\]\[\]=2670999714&filter\[id\]\[in\]\[\]=2670999768](https://api.smsclub.mobi/rcs?filter[id][in][]=2670999714&filter[id][in][]=2670999768)
- Якщо необхідно отримати інформацію по списку sms повідомлень без деталізації по id <https://api.octotoo.ua/rcs>

Варіанти відповіді (Responses)

Код (status)	Опис	Приклад
200	Успішне надсилання RCS -повідомлення	<pre>{ "status": "success", "data": [{ "id": 2442222222, "state": 0, "send_date": "2024-02-27T12:56:17.000Z", "phone": 380961111111, "price": "1.00000", "MCC": 255, "MNC": 6, "count_sms": 1, "currency": "UAH" }] }</pre>
404	Повідомлення не знайдено	<pre>{ "name": "Not Found", "message": "Not Found", "code": 0, "status": 404 }</pre>

Опис отриманих даних на **GET** запит по конкретному ID RCS повідомлення

Отримані дані	Опис
<pre>{ "status": "success",</pre>	Статус запиту, в даному випадку "успішний"
<pre>"data": [</pre>	Далі відображаються деталі по кожному повідомленню
<pre>"id": 2442222222,</pre>	ID повідомлення в системі



"state": 0,	Статус повідомлення "state": 0 - відправлено "state": 1, - доставлено "state": 2, - не доставлено "state": 3, - термін доставки пройшов
"send_date": "2024-02-27T12:56:17.000Z",	Дата відправки повідомлення
"phone": 380961111111,	Номер телефону, на який відправлене повідомлення
"price": "1.00000",	Ціна за повідомлення
"MCC": 255, "MNC": 6,	Напрямок країни та оператора, на який було відправлене повідомлення
"count_sms": 1,	Кількість повідомлень
"currency": "UAH",	Валюта, в якій сплачується плата за повідомлення (price)



6. Каскадні розсилки (cascade)

Функціонал дає змогу зменшити витрати на розсилки та підвищити ефективність комунікацій завдяки гнучким налаштуванням cascades.

Користувачі можуть обирати канали (Viber, RCS, SMS), налаштовувати їхню послідовність і запускати автоматизовану доставку повідомлень за заздалегідь визначеним сценарієм.

6.1 Відправка каскадного повідомлення

Необхідно виконати **POST** запит на url `https://api.octotoo.ua/cascades`

Токен для API підключення ви знайдете в особистому кабінеті в Профіль користувача (інструкція по [лінку](#))

6.1.1. Приклад json запиту по відправці каскаду

```
{
  "channels": [
    "rcs",
    "sms",
    "viber"
  ],
  "recipients": [
    {
      "phone": "380961111111"
    },
    {
      "phone": "380962222222"
    }
  ],
  "rcs": {
    "from": "Sender",
    "text": "Message text",
    "buttonText": "Button text",
    "buttonUrl": "https://example.com",
    "imgUrl": "https://example.com/image.jpg",
    "rcsTtl": 120
  },
  "sms": {
    "from": "Sender",
    "text": "Message text",
    "lifetime": 180
  },
  "viber": {
    "from": "ViberSender",
    "text": "Message viber text",
    "lifetime": 180
  },
  "nameDelivery": "string",
  "sendFromDate": "2025-05-15 12:00:00"
}
```



6.1.2. Блок channels

{ "channels": [	Об'єкт, що містить дані для формування послідовності відправки -повідомлення.		
Параметр	Формат даних	Обов'язково/ Опціонально	Опис поля
channels*	channels Channels[string xml: OrderedMap { "name": "item" } xml: name: item Enum: [sms, rcs]	Обов'язково	Перерахунок каналів відправки, тобто в якій послідовності потрібно відправляти повідомлення. Має бути не менше 2 назв rcs та/або sms та/або viber.

6.1.3 Блок recipients

описаний у розділі [3.1.3 Блок recipients](#)

6.1.4. Блок rcs

описаний у розділі [5.1 Відправка RCS повідомлення](#). Можна вказати всі значення, які передбачені в повідомленні типу RCS

6.1.5. Блок sms

описаний у розділі [2.1 Відправка SMS повідомлення по API](#). Можна вказати всі значення, які передбачені в повідомленні типу SMS

6.1.6 Блок viber

описаний у розділі [3.1 Відправка Viber повідомлення по API](#). Можна вказати всі значення, які передбачені в повідомленні типу Viber

6.2. Отримати статус по каскадній розсилці

Необхідно виконати **GET** запит на url <https://api.octotoo.ua/cascades/{cascadeld}>

Токен для API підключення ви знайдете в особистому кабінеті в Профіль користувача (інструкція по [лінку](#))

Варіанти відповіді (Responses)

Код (status)	Опис	Приклад
200	Успішна відповідь	{

		<pre> "status": "success", "data": { "id": 1234, "distribution_id": "0197162d-87d4-7a65-bed7-c12eff34dfe4", "channels": ["sms", "rcs"], "sequences": [{ "recipient": { "phone": "380961111111" }, "sms": { "state": "not_used" }, "rcs": { "state": "not_used" } }, { "recipient": { "phone": "380962222222" }, "sms": { "state": "not_used" }, "rcs": { "state": "not_used" } }] } </pre>
404	Повідомлення не знайдено	<pre> { "name": "Not Found", "message": "Not Found", "code": 0, "status": 404 } </pre>
500 Internal server error	Внутрішня помилка сервера	



7. Реєстрація Альфа - імені по запиту

Функціонал дає змогу зареєструвати Альфа ім'я (далі - AI) виконавши запит по API.

Необхідно виконати **POST** запит на url <https://api.octotoo.ua/originators/registration-originators>
Токен для API підключення ви знайдете в особистому кабінеті в Профіль користувача (інструкція по [лінку](#))

7.1 Приклад json запиту по відправці запиту на створення AI

```
{
  "company": "string",
  "site": "string",
  "originator": "string",
  "description": "string",
  "subject": "string",
  "company_type": "string",
  "type": "viber"
}
```

RegistrationOriginators{			
Об'єкт, що містить дані для формування запиту на реєстрацію Альфа імені			
Параметр	Формат даних	Обов'язково/ Опціонально	Опис поля
company*	string	Обов'язково	Юридична назва компанії
site*	string	Обов'язково	URL-адреса сайту або посилання на компанію в соціальних мережах
originator*	string	Обов'язково	Альфа ім'я
description*	string	Обов'язково	Опис послуг компанії
subject*	string	Обов'язково	Мета інформування
company_type*	string	Обов'язково	Тип компанії
type*	strings Enum: [sms, viber, rcs]	Обов'язково	Вказати для якого типу повідомлень потрібно створити Альфа ім'я sms або viber або rcs. На кожен тип потрібно формувати окремий запит.
okpo	integer	Опціонально	ЄДРПОУ організації
inn	integer	Опціонально	Реєстраційний номер платника податків



7.2 Коди відповіді при створенні AI

Код відповіді	Опис	Приклад відповіді json
201 Successful request	Успішна відповідь, запит на реєстрацію AI прийнято	<pre>{ "successful_request": { "originator": "NewOriginator", "state": "moderation" }}</pre>
400 Registration error	Неуспішна відповідь	<pre>{ "error_request": { "originator": "Alpha Name cannot be blank." }}</pre>