

Ocean Media Lead Platform

Client Ingest API — dokumentacja integracyjna

Wersja dokumentu: v1.1

Wersja API: v1.0

16 sierpnia 2026

Dokument dla systemów klienta, które opcjonalnie przekazują pełne dane leadów Meta do Ocean Media Lead Platform. Standardowe połączenie Meta nie wymaga korzystania z Client Ingest API.

Klucz API jest przekazywany klientowi osobnym, bezpiecznym kanałem i nie powinien być wpisywany do tej dokumentacji. Kontakt techniczny i organizacyjny: biuro@ocean-media.pl.

1. Cel i zakres integracji

Client Ingest API jest dodatkowym kanałem integracyjnym służącym do bezpośredniego przekazywania do Ocean Media Lead Platform pełnego obiektu JSON leada pozyskanego z Meta lub systemu klienta. System klienta wysyła każdy nowy lead przez HTTPS. Platforma zapisuje kompletny logiczny obiekt JSON w PostgreSQL JSONB oraz wyodrębnia dane techniczne potrzebne do identyfikacji, deduplikacji i raportowania.

- Client Ingest jest kanałem opcjonalnym. Gdy klient podłączy zasoby Meta bezpośrednio do platformy, źródłem referencyjnym dla leadów Meta są dane pozyskane bezpośrednio z Meta przez webhook i Graph API.
- Payload Client Ingest nie jest ograniczony do telefonu, imienia, nazwiska ani e-maila — może zawierać wszystkie pola przekazywane przez system klienta.
- Dane zapisane dziś mogą być wykorzystane w przyszłych raportach, o ile dane pole znajdowało się w pierwotnym payloadzie.
- Jeden rzeczywisty Meta lead ma jeden rekord w bazie, nawet jeśli ten sam lead zostanie zaobserwowany również przez bezpośredni webhook Meta, odczyt Graph API albo proces reconciliation.

Uwaga o JSONB: zachowywana jest pełna struktura i wartości JSON, ale JSONB nie gwarantuje zachowania oryginalnej kolejności kluczy, odstępów ani formatowania tekstowego. Do raportowania i analizy nie ma to znaczenia.

2. Dane połączeniowe

Base URL	https://verify.ocean-media.pl
Endpoint	POST /api/v1/leads/meta
Pełny URL	https://verify.ocean-media.pl/api/v1/leads/meta
Uwierzytelnienie	Nagłówek X-API-Key
Content-Type	application/json
Transport	HTTPS
Kodowanie	UTF-8

Klucz API: każda integracja powinna używać własnego, dedykowanego klucza. Klucza nie należy umieszczać w URL, logach aplikacyjnych ani publicznych repozytoriach.

3. Kontrakt payloadu

Body żądania musi być obiektem JSON. Poza identyfikatorem Meta Lead ID platforma nie narzuca sztywnego zestawu pól.

3.1. Wymagany identyfikator leada Meta

Platforma rozpoznaje rzeczywisty Meta Lead ID w jednej z następujących postaci:

- leadgen_id — zalecane przy własnym wrapperze JSON klienta;
- lead_id — akceptowany alias;

- `id` — naturalny wariant przy przekazywaniu kompletnego, natywnego obiektu leada pobranego z Meta Graph API.

Krytyczne: wartość ta musi być rzeczywistym Meta `leadgen_id`. Wewnętrzny identyfikator CRM klienta powinien być przesłany jako osobne pole, np. `crm_id`. Dzięki temu Client Ingest oraz bezpośrednie kanały Meta mogą zostać automatycznie rozpoznane jako ten sam lead.

3.2. Zalecany wariant — natywny pełny obiekt Meta

```
{
  "id": "123456789012345",
  "created_time": "2026-08-15T18:45:10+0000",
  "ad_id": "987654321001",
  "form_id": "987654321002",
  "campaign_id": "987654321003",
  "field_data": [
    {
      "name": "full_name",
      "values": ["Anna Nowak"]
    },
    {
      "name": "phone_number",
      "values": ["+48555111222"]
    },
    {
      "name": "email",
      "values": ["anna.nowak@example.pl"]
    }
  ]
}
```

3.3. Wariant z własnym wrapperem klienta

```
{
  "leadgen_id": "123456789012345",
  "crm_id": "CRM-2026-008812",
  "created_time": "2026-08-15T18:45:10+0000",
  "source_system": "crm-client",
  "field_data": [
    {
      "name": "full_name",
      "values": ["Anna Nowak"]
    },
    {
      "name": "phone_number",
      "values": ["+48555111222"]
    }
  ],
  "custom": {
    "region": "wielkopolskie",
    "sales_channel": "facebook"
  }
}
```

Własne pola, zagnieżdżone obiekty i tablice są dopuszczalne. Zalecenie: nie usuwać z payloadu pól tylko dlatego, że aktualny raport ich jeszcze nie wykorzystuje.

4. Przykład wywołania cURL

```
curl -i -sS \
  -X POST \
  "https://verify.ocean-media.pl/api/v1/leads/meta" \
  -H "X-API-Key: <TENANT_API_KEY>" \
  -H "Content-Type: application/json" \
  --data-binary @lead.json
```

Klucz API należy podstawić w nagłówku. Plik `lead.json` powinien zawierać kompletny JSON leada.

5. Odpowiedź API i idempotencja

Każda poprawna odpowiedź zawiera nagłówek X-Lead-Ingest-Result. Jest on najprostszym sposobem rozpoznania, co stało się z żądaniem.

HTTP	X-Lead-Ingest-Result	Znaczenie	Co powinien zrobić klient
201	created	Nowy lead został zapisany.	Uznać wysyłkę za zakończoną.
200	duplicate	Identyczny lead był już zapisany.	Uznać wysyłkę za zakończoną. To poprawny wynik retry.
200	enriched	Wcześniej istniał częściowy lead z webhooka Meta; pełny payload klienta uzupełnił ten sam rekord przed potwierdzeniem Graph API.	Uznać wysyłkę za zakończoną.
200	observed	Lead już istniał, a żądanie nie pogorszyło jego danych; platforma odnotowała kolejną ścieżkę dostarczenia.	Uznać wysyłkę za zakończoną.
401	—	Brak, nieprawidłowy lub unieważniony klucz API.	Nie ponawiać automatycznie bez poprawienia klucza.
409	—	Ten sam Meta Lead ID ma już inne kompletne dane.	Zatrzymać retry i wyjaśnić rozbieżność.
422	—	Nieprawidłowy JSON lub brak rozpoznawalnego Meta Lead ID.	Poprawić payload; nie ponawiać bez zmian.
5xx	—	Błąd chwilowy po stronie usługi lub infrastruktury.	Ponowić dokładnie to samo żądanie.

5.1. Przykładowa odpowiedź po utworzeniu leada

```
HTTP/1.1 201 Created
X-Lead-Ingest-Result: created
Content-Type: application/json
```

```
{
```

```
{
  "id": "5a8c6e75-3d1e-47ee-8dd3-0fc4fa61aa35",
  "tenant_id": "29d36a9d-3c9d-4ff0-bc65-71a9f1fa22c5",
  "source": "meta",
  "source_lead_id": "123456789012345",
  "payload": { "...": "pełny przesłany JSON" },
  "payload_hash": "...",
  "data_status": "complete",
  "ingestion_sources": ["client_ingest"],
  "meta_verified_at": null,
  "occurred_at": "2026-08-15T20:45:10+02:00",
  "received_at": "2026-08-15T20:45:11.315000+02:00",
  "updated_at": "2026-08-15T20:45:11.315000+02:00"
}
```

5.2. Zasada deduplikacji

Kluczem logicznym leada jest połączenie: tenant + source=meta + Meta Lead ID. W praktyce oznacza to:

1. Pierwsze wysłanie pełnego leada → 201 created.
2. Ponowienie dokładnie tego samego pełnego payloadu → 200 duplicate; nie powstaje drugi rekord i licznik leadów nie rośnie.
3. Jeżeli ten lead wcześniej dotarł jako częściowy webhook Meta i nie został jeszcze potwierdzony przez Graph API, pełny Client Ingest może uzupełnić ten sam rekord i zwrócić 200 enriched.
4. Jeżeli rekord został już potwierdzony bezpośrednio przez Meta Graph API, dane z Client Ingest nie nadpisują autorytatywnego payloadu Meta. Rozbieżność wymaga diagnostyki zamiast cichego zastąpienia danych.

6. Zalecana polityka retry

Client Ingest został zaprojektowany tak, aby bezpiecznie obsługiwać ponowienia po utracie połączenia lub braku odpowiedzi HTTP.

- Przy błędzie sieciowym lub odpowiedzi 5xx ponowić dokładnie to samo żądanie: ten sam Meta Lead ID i ten sam payload.
- Dopuszczalny jest exponential backoff po stronie klienta, np. 1 s, 5 s, 15 s, 60 s, a następnie rzadsze próby zgodnie z polityką systemu klienta.
- 200 duplicate jest sukcesem — oznacza, że poprzednia próba została już zapisana.
- 401, 409 i 422 wymagają korekty danych lub konfiguracji i nie powinny być bez końca ponawiane automatycznie.

Przykład: serwer zapisze lead, ale połączenie zerwie się przed odebraniem 201 przez klienta. Klient ponawia identyczne żądanie i otrzymuje 200 duplicate. W bazie nadal istnieje jeden lead.

7. Przykład integracji PHP

```
<?php
$apiUrl = 'https://verify.ocean-media.pl/api/v1/leads/meta';
$apiKey = getenv('OCEAN_MEDIA_LEAD_API_KEY');

$lead = [
    'leadgen_id' => '123456789012345',
    'crm_id' => 'CRM-2026-008812',
    'created_time' => '2026-08-15T18:45:10+0000',
    'field_data' => [
        ['name' => 'full_name', 'values' => ['Anna Nowak']],
        ['name' => 'phone_number', 'values' => ['+48555111222']],
    ],
];

$ch = curl_init($apiUrl);
curl_setopt_array($ch, [
    CURLOPT_POST => true,
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_HEADER => true,
    CURLOPT_HTTPHEADER => [
        'X-API-Key: ' . $apiKey,
        'Content-Type: application/json',
    ],
    CURLOPT_POSTFIELDS => json_encode(
        $lead,
        JSON_UNESCAPED_UNICODE | JSON_UNESCAPED_SLASHES
    ),
    CURLOPT_TIMEOUT => 15,
]);

$response = curl_exec($ch);
$httpCode = curl_getinfo($ch, CURLINFO_HTTP_CODE);

if ($response === false) {
    throw new RuntimeException(curl_error($ch));
}

curl_close($ch);

if (!in_array($httpCode, [200, 201], true)) {
    throw new RuntimeException('Lead API HTTP ' . $httpCode);
}
```

Klucz powinien być przechowywany poza kodem źródłowym, np. w zmiennej środowiskowej lub bezpiecznym magazynie sekretów.

8. Przykład integracji C# / .NET

```
using System.Net.Http.Json;

var apiKey = Environment.GetEnvironmentVariable(
    "OCEAN_MEDIA_LEAD_API_KEY"
) ?? throw new InvalidOperationException("Missing API key");

using var http = new HttpClient
{
    BaseAddress = new Uri("https://verify.ocean-media.pl/")
};

http.DefaultRequestHeaders.Add("X-API-Key", apiKey);

var lead = new
{
    leadgen_id = "123456789012345",
    crm_id = "CRM-2026-008812",
    created_time = "2026-08-15T18:45:10+0000",
    field_data = new object[]
    {
        new { name = "full_name", values = new[] { "Anna Nowak" } },
        new { name = "phone_number", values = new[] { "+48555111222" } }
    }
};

using var response = await http.PostAsJsonAsync(
    "api/v1/leads/meta",
    lead
);

var ingestResult = response.Headers.TryGetValues(
    "X-Lead-Ingest-Result",
    out var values
) ? values.FirstOrDefault() : null;

if (response.IsSuccessStatusCode)
{
    var body = await response.Content.ReadAsStringAsync();
    Console.WriteLine($"{ingestResult}: {body}");
}
else
{
    var body = await response.Content.ReadAsStringAsync();
    throw new HttpRequestException(
        $"Lead API {(int)response.StatusCode}: {body}"
    );
}
```

9. Odczyt i statystyki

Ten sam klucz klienta może zostać użyty do odczytu wyłącznie danych własnego tenanta. Znajomość UUID leada innego klienta nie daje do niego dostępu. Parametr meta_verified pozwala rozróżnić leady potwierdzone bezpośrednio przez Meta Graph API od rekordów jeszcze niepotwierdzonych w Meta.

Lista leadów	GET /api/v1/leads?source=meta&limit=100&offset=0
Tylko kompletne	GET /api/v1/leads?source=meta&data_status=complete
Statystyka	GET /api/v1/leads/stats?source=meta&data_status=complete
Zakres czasu	GET /api/v1/leads/stats? source=meta&date_from=<ISO8601>&date_to=<ISO8601>

Pojedynczy lead

GET /api/v1/leads/{lead_uuid}

**Potwierdzone
bezpośrednio przez Meta**

GET /api/v1/leads?source=meta&meta_verified=true

Zakres czasu: raportowanie wykorzystuje occurred_at, jeśli czas powstania leada został rozpoznany w payloadzie; w przeciwnym razie używany jest received_at. Parametr date_to jest traktowany jako górna granica przedziału.

10. Współpraca z bezpośrednim Meta Webhook / Graph API

Client Ingest może działać samodzielnie lub równolegle z bezpośrednią integracją Meta. Przy aktywnym podłączeniu Meta platforma odbiera zdarzenia leadgen w czasie rzeczywistym przez webhook, następnie pobiera pełny lead z Meta Graph API. Niezależny proces reconciliation okresowo odczytuje formularze i leady z Graph API, aby wykrywać i uzupełniać ewentualne braki.

Przykładowy przebieg:

1. Meta webhook: leadgen_id = 12345 -> rekord partial
2. Meta Graph API: id = 12345 -> ten sam rekord, complete, meta_verified_at ustawione
3. Reconciliation: id = 12345 -> potwierdzenie obecności / odświeżenie danych autorytatywnych
4. Client Ingest: leadgen_id = 12345 -> ten sam rekord; nie zastępuje danych potwierdzonych przez Meta

Dlatego nie należy zastępować Meta Lead ID identyfikatorem CRM. Pole crm_id można przechowywać dodatkowo w payloadzie. Dla raportowania liczby leadów Meta źródłem referencyjnym są rekordy potwierdzone bezpośrednio przez Meta, gdy bezpośrednia integracja Meta jest aktywna.

11. Wymagania bezpieczeństwa

- Używać wyłącznie HTTPS.
- Klucz API przechowywać jak hasło; nie umieszczać w URL ani w repozytorium.
- Nie logować nagłówka X-API-Key.
- Dla każdej niezależnej integracji używać osobnego klucza, aby można było ją niezależnie unieważnić.
- W przypadku podejrzenia ujawnienia klucza niezwłocznie zgłosić potrzebę jego unieważnienia i wydania nowego.
- Przesyłać wyłącznie dane, do których klient posiada podstawę i uprawnienie do przetwarzania. Platforma archiwizuje pełny otrzymany payload, dlatego zakres danych powinien być świadomą decyzją klienta.

12. Procedura uruchomienia klienta

1. Ocean Media tworzy tenant klienta i dedykowany klucz API dla konkretnej integracji Client Ingest.
2. Klucz API jest przekazywany osobnym bezpiecznym kanałem.
3. Klient konfiguruje POST HTTPS na /api/v1/leads/meta i wysyła pełny JSON leada.
4. Wspólnie wykonujemy test z unikalnym testowym Meta Lead ID.
5. Oczekiwany pierwszy wynik: HTTP 201 + X-Lead-Ingest-Result: created.
6. Klient wysyła identyczny payload drugi raz.
7. Oczekiwany wynik retry: HTTP 200 + X-Lead-Ingest-Result: duplicate.
8. Po potwierdzeniu testu integracja zostaje przełączona na dane produkcyjne. Jeżeli klient korzysta również z bezpośredniego podłączenia Meta, oba kanały mogą działać równolegle z deduplikacją po Meta Lead ID.

13. Szybka karta referencyjna

POST	<code>https://verify.ocean-media.pl/api/v1/leads/meta</code>
Nagłówek 1	<code>X-API-Key: <dedykowany klucz klienta></code>
Nagłówek 2	<code>Content-Type: application/json</code>
Body	Dowolny pełny obiekt JSON leada Meta + rzeczywisty Meta Lead ID
Sukces nowy	201 / created
Sukces retry	200 / duplicate
Uzupełnienie partial	200 / enriched
Konflikt	409 — ten sam Meta Lead ID, inne kompletne dane
Walidacja	422 — niepoprawny JSON lub brak Meta Lead ID
Autoryzacja	401 — nieprawidłowy lub nieaktywny klucz API
Rola kanału	Opcjonalny kanał dodatkowy; przy bezpośrednim podłączeniu Meta źródłem referencyjnym jest Meta Graph API