WAGO Payment ID

Developer Integration Guide & Security Reference

Edisi Terpadu: v2.4 (Unified Edition)

Dokumen ini ditujukan bagi Developer dan Administrator Sistem yang berintegrasi dengan WAGO Payment ID. Panduan ini mencakup alur pembuatan transaksi dasar hingga implementasi keamanan tingkat lanjut (*Server-Side Redaction* & *HMAC Anti-Replay*).

1. Pendahuluan & Alur Integrasi

WAGO Payment ID adalah sistem *Payment Gateway* yang menjembatani transaksi dari website merchant (klien) ke sistem pembayaran QRIS Dinamis.

Flow Transaksi Standar:

1. **Inisiasi (POST):** Backend website Anda menembak API WAGO untuk membuat pesanan baru.
2. **Redirect:** Anda mengarahkan pembeli ke URL Checkout WAGO yang dikembalikan oleh API.
3. **Verifikasi (Callback/Webhook):** Saat pembeli selesai membayar atau membatalkan, WAGO akan mengembalikan pembeli ke website Anda via `callback_url` beserta **Digital Signature (HMAC)**.
4. **Validasi & Rilis:** Backend Anda memvalidasi Signature tersebut, lalu mengeksekusi logika bisnis (merilis layanan/produk).

2. Otentikasi & Keamanan Dasar

Semua permintaan API (POST/PUT/GET) dari server Anda ke WAGO wajib menyertakan otentikasi berikut di dalam *Headers*:

- `x-api-key`: Berisi **Password Merchant** yang diberikan oleh Admin WAGO.

Peringatan Keamanan: Panggilan API dan penyimpanan *Secret Key* harus selalu dilakukan dari sisi server (Backend seperti Node.js, PHP, Python). **Jangan pernah** meletakkan `x-api-key` atau *Secret Key* di dalam kode Frontend (HTML/JavaScript Browser) yang bisa dilihat oleh publik.

3. Membuat Transaksi Baru (POST)

Gunakan endpoint ini untuk menginisiasi pembayaran baru sebelum mengarahkan user ke halaman WAGO.

Endpoint: `POST https://wago-payment-id.vercel.app/api/order`

3.1 Parameter Request (Body JSON)

Parameter	Wajib	Deskripsi
`order_id`	Ya	ID unik pesanan dari sistem Anda (Contoh: TX-1001).
`app_id`	Ya	Nama/ID bisnis Anda yang terdaftar di WAGO.
`nominal`	Ya	Harga total produk (Angka murni, tanpa titik/koma).
`callback_url`	Ya	URL tujuan di website Anda jika transaksi selesai/batal. **URL ini harus sesuai dengan Domain Whitelist yang terdaftar.**
`customer_name`	Tidak	Nama pelanggan pembeli produk.
`customer_email`	Tidak	Email pelanggan untuk pengiriman notifikasi.
`product_details`	Tidak	Array objek detail produk (`name` & `price`) agar struk menampilkan rincian produk.

3.2 Contoh Payload & Response

JSON Request Payload

```
{
  "order_id": "3DQRQV921X",
  "app_id": "CINELIST_ELITE",
  "customer_name": "NAMA PELANGGAN",
  "customer_email": "customer@email.com",
  "nominal": 70000,
  "callback_url": "https://website-anda.com/payment/verify",
  "product_details": [
    {
      "name": "Tiket Premier Studio 1",
      "price": 70000
    }
  ]
}
```

Response Sukses (201 Created): API akan mengembalikan data JSON. Ambil nilai `order_id`, lalu arahkan pengguna Anda ke URL Checkout.

4. Definisi Status & Aksi Website Merchant

Saat transaksi berubah status, WAGO akan mengirimkan data ke `callback_url` Anda. Sesuaikan logika bisnis Anda dengan status berikut:

Status	Kondisi di Sistem WAGO	Aksi di Sistem Merchant
PENDING	Menunggu pembayaran QRIS / Waktu belum habis.	Tampilkan pesan "Menunggu Pembayaran". Jangan rilis produk.
SUCCESS	Pembayaran terkonfirmasi lunas oleh sistem.	Update database Anda menjadi Lunas, dan rilis produk/layanan.
CANCELED	User membatalkan atau waktu kedaluwarsa.	Ubah status menjadi Batal/Gagal. Kembalikan stok barang jika ada.

5. Verifikasi Keamanan Lanjutan (KRUSIAL)

> **PENTING: Anti-Fraud & Anti-Replay**
>
> Untuk mencegah Fraud (penipuan URL callback), WAGO menerapkan **HMAC SHA-256 Signature** dan **Anti-Replay Timestamp**. Anda **WAJIB** memvalidasi Signature ini sebelum merilis produk.

5.1 Parameter Callback & Rumus Signature

WAGO akan menyisipkan parameter berikut pada URL Query String atau Webhook Body: `order_id`, `status`, `nominal`, `t` (Unix Timestamp), dan `sig` (HMAC SHA-256).

$$Payload = order\_id + ":" + status + ":" + nominal + ":" + t$$

5.2 Contoh Implementasi JavaScript (Node.js)

JavaScript (Node.js)

```
const crypto = require('crypto');

app.get('/payment/verify', (req, res) => {
  const { order_id, status, nominal, t, sig } = req.query;
  const SECRET = process.env.WAGO_CALLBACK_SECRET;

  // 1. Cek Waktu DULU (Toleransi maksimal 5 menit / 300 detik)
  const now = Math.floor(Date.now() / 1000);
  if (Math.abs(now - parseInt(t)) > 300) return res.status(403).send("Request Expired");

  // 2. Hitung Hash dari payload yang diterima
  const rawPayload = `${order_id}:${status}:${nominal}:${t}`;
  const expectedSig = crypto.createHmac('sha256',
SECRET).update(rawPayload).digest('hex');

  // 3. Bandingkan secara aman
  if (!crypto.timingSafeEqual(Buffer.from(sig), Buffer.from(expectedSig))) {
    return res.status(403).send("Invalid Signature");
  }

  // 4. Lolos Validasi: Rilis produk
  if (status === 'SUCCESS') {
      // Update DB & Berikan akses layanan
  }
  res.send("Proses Selesai");
});
```

5.3 Contoh Implementasi PHP

PHP

```php
<?php
$secret = "WAGO_CALLBACK_SECRET_ANDA";
$t = $_GET['t'];
$sig = $_GET['sig'];

// 1. Cek batas waktu TERLEBIH DAHULU untuk efisiensi
if (abs(time() - intval($t)) > 300) {
    http_response_code(403);
    die("Request Expired");
}

// 2. Lanjut komputasi Hash
$payload = "{$_GET['order_id']}:{$_GET['status']}:{$_GET['nominal']}:{$t}";
$expected = hash_hmac('sha256', $payload, $secret);

// 3. Validasi dengan hash_equals
if (hash_equals($expected, $sig)) {
    if ($_GET['status'] === 'SUCCESS') {
        // Rilis Produk
    }
    echo "SUCCESS";
} else {
    http_response_code(403);
    die("Invalid Signature");
}
?>
```

6. Pengambilan Data & Privasi (Data Redaction)

Jika sewaktu-waktu sistem Anda perlu menarik ulang data transaksi spesifik, gunakan endpoint: `GET /api/order?id=[ORDER_ID]`

Kebijakan Privasi (Server-Side Redaction):

- Jika endpoint ini dipanggil **tanpa** menyertakan header `x-api-key` yang valid, API akan **MENGHAPUS** field sensitif dari JSON response.
- Field yang dihapus: `customer_name`, `customer_email`, dan `callback_url`. Data tersebut tidak akan muncul atau bernilai null, melainkan hilang sepenuhnya dari struktur JSON.

- Untuk mendapatkan data utuh, pastikan server Anda selalu menyisipkan `x-api-key`.