

API Documentation

Payment Gateway QRIS — OrderKuota Un-Official API Client v1.0

Base URL

Semua endpoint menggunakan prefix `/api` . Base URL: `https://orkut.nickystore.my.id`

Contoh: `https://orkut.nickystore.my.id/api/payment/create`

Autentikasi

Ada dua metode autentikasi:

1. Cookie-Based (Web Dashboard)

Login via `POST /api/login` mengembalikan httpOnly cookie `token` . Digunakan untuk akses dashboard dan endpoint user.

2. API Key (Bot / Aplikasi Eksternal)

Gunakan header `x-api-key` untuk autentikasi bot. API key didapat dari Dashboard → Akun → API Key.

Master key default: `paymet-bot-2026` (bisa diubah via env `BOT_KEY`).

`GET` `/api/session`

```
{ "userId": 123456, "username": "john", "role": "free", "api_key": "...",  
  "payment_count": 3, "usage_limit": 5, "total_income": 150000 }
```

`POST` `/api/logout`

```
{ "success": true }
```

`POST` `/api/user/api-key`

```
{ "success": true, "api_key": "new-api-key-here" }
```

Info: Regenerate API key akan membuat key lama tidak berlaku.

Error Handling

Semua endpoint mengembalikan error dalam format JSON:

```
{ "error": "Pesan error" }
```

Kode HTTP

400	Bad Request — input tidak valid
401	Unauthorized — token/API key tidak valid
403	Forbidden — banned, limit habis, atau bukan owner
404	Not Found — resource tidak ditemukan
409	Conflict — akun OrderKuota sudah terpakai
429	Too Many Requests — rate limit exceeded
500	Internal Server Error
503	Service Unavailable — env var kurang

Tips: Selalu cek `res.ok` pada response fetch. Kalau `!res.ok`, baca `res.json().error` untuk pesan error.

Panduan Integrasi Bot WhatsApp

Flow integrasi untuk bot WhatsApp atau aplikasi eksternal:

1. `POST /api/bot/payment/create` dengan header `x-api-key: YOUR_API_KEY` → dapat `id` + `qr_image_url`
2. Kirim QR image ke customer via WhatsApp
3. Tiap 5-10 detik `GET /api/bot/payment/:id` → cek status payment
4. Kalau `status === "succeed"` → kirim notifikasi "Pembayaran berhasil!"
5. Kalau `status === "cancelled"` → "Pembayaran dibatalkan"
6. Kalau masih `status === "pending"` → "Masih menunggu pembayaran"

Contoh Kode (JavaScript)

```
// Buat payment
const create = await
fetch('https://orkut.nickystore.my.id/api/bot/payment/create', { method: 'POST',
headers: { 'Content-Type': 'application/json', 'x-api-key': 'YOUR_KEY' }, body:
JSON.stringify({ amount: 10000 }) });
const data = await create.json();
console.log(data.payment.qr_image_url); // Kirim ini ke customer // Cek status
(loop tiap 5 detik)
setInterval(async () => {
  const res = await
  fetch(`https://orkut.nickystore.my.id/api/bot/payment/${paymentId}`, { headers: {
    'x-api-key': 'YOUR_KEY' } });
  const d = await res.json();
  if (d.payment.status === 'succeed') {
    console.log('Pembayaran berhasil!');
    clearInterval(interval);
  },
  5000);
```

Contoh dengan Python

```
import requests, time # Buat payment
r = requests.post('https://orkut.nickystore.my.id/api/bot/payment/create', json={
  'amount': 10000}, headers={'x-api-key': 'YOUR_KEY'})
payment = r.json()
['payment']
print(payment['qr_image_url']) # Kirim QR ke customer # Polling status
while True:
  r = requests.get(f'https://orkut.nickystore.my.id/api/bot/payment/{payment["id"]}',
  headers={'x-api-key': 'YOUR_KEY'})
  status = r.json()['payment']['status']
  if status == 'succeed':
    print('Berhasil!')
    break
  elif status == 'cancelled':
    print('Dibatalkan')
    break
  time.sleep(5)
```

Catatan: Pastikan sudah menghubungkan akun OrderKuota di dashboard dan menyimpan QRIS string sebelum menggunakan payment endpoint. Limit gratis:

5
pembayaran/hari

Contoh Implementasi API

Berikut contoh lengkap penggunaan endpoint utama menggunakan **Python**, **JavaScript**, dan **cURL**.

Base URL: <https://orkut.nickystore.my.id> . Gunakan token autentikasi dashboard via cookie atau API Key via header **x-api-key** .

1. Cek Saldo Utama

Mendapatkan saldo utama akun OrderKuota.

GET <https://orkut.nickystore.my.id/api/orkut/qrisk>

Parameter Tidak ada parameter

Auth Cookie session (login dashboard)

```
{ "success": true, "qrisk_url": "https://...", "qrisk_name": "NAMA TOKO",  
  "qrisk_balance": 500000, "balance": 100000, "qrisk_string": "..." }
```

Python

```
import requests url = "https://orkut.nickystore.my.id/api/orkut/qrisk" headers = {  
  "Cookie": "token=TOKEN_DASHBOARD" } r = requests.get(url, headers=headers) data =  
r.json() print("Saldo Utama:", data.get("balance")) print("Saldo QRIS:",  
data.get("qrisk_balance"))
```

JavaScript (fetch)

```
const res = await fetch("https://orkut.nickystore.my.id/api/orkut/qrisk", {  
  credentials: "include" }); const data = await res.json(); console.log("Saldo  
Utama:", data.balance); console.log("Saldo QRIS:", data.qrisk_balance);
```

cURL

```
curl -X GET "https://orkut.nickystore.my.id/api/orkut/qrisk" \ -H "Cookie: token=TOKEN_DASHBOARD"
```

2. Cek Saldo QRIS

Mendapatkan saldo QRIS akun OrderKuota (endpoint sama dengan Saldo Utama, field berbeda).

```
GET https://orkut.nickystore.my.id/api/orkut/qrisk
```

Parameter	Tidak ada parameter
Auth	Cookie session (login dashboard)

```
{ "success": true, "qrisk_balance": 500000, "balance": 100000, ... }
```

Python

```
import requests url = "https://orkut.nickystore.my.id/api/orkut/qrisk" headers = { "Cookie": "token=TOKEN_DASHBOARD" } r = requests.get(url, headers=headers) data = r.json() print("Saldo QRIS:", data.get("qrisk_balance"))
```

JavaScript (fetch)

```
const res = await fetch("https://orkut.nickystore.my.id/api/orkut/qrisk", { credentials: "include" }); const data = await res.json(); console.log("Saldo QRIS:", data.qrisk_balance);
```

cURL

```
curl -X GET "https://orkut.nickystore.my.id/api/orkut/qrisk" \ -H "Cookie: token=TOKEN_DASHBOARD"
```

3. Buat Payment QRIS

Membuat payment baru dan mendapatkan QRIS image untuk dibayar customer.

```
POST https://orkut.nickystore.my.id/api/payment/create
```

Parameter	Tipe	Deskripsi
<code>amount</code>	number	Nominal pembayaran (kelipatan Rp 1.000)
Auth	Cookie session (dashboard) atau header <code>x-api-key</code>	

Request:

```
{ "amount": 10000 } Response:
{ "success": true, "payment": { "id": "pay_abc123", "amount": 10000, "status":
"pending", "qr_image_url": "https://...", "balance_before": 500000 } }
```

Python

```
import requests url = "https://orkut.nickystore.my.id/api/payment/create" headers
= { "x-api-key": "API_KEY_ANDA", "Content-Type": "application/json" } body = {
"amount": 10000 } r = requests.post(url, json=body, headers=headers) payment =
r.json()["payment"] print("Payment ID:", payment["id"]) print("QR Image:",
payment["qr_image_url"])
```

JavaScript (fetch)

```
const res = await fetch("https://orkut.nickystore.my.id/api/payment/create", {
method: "POST", headers: { "Content-Type": "application/json", "x-api-key":
"API_KEY_ANDA" }, body: JSON.stringify({ amount: 10000 }) }); const data = await
res.json(); console.log("Payment ID:", data.payment.id); console.log("QR Image:",
data.payment.qr_image_url);
```

cURL

```
curl -X POST "https://orkut.nickystore.my.id/api/payment/create" \ -H "x-api-key:
API_KEY_ANDA" \ -H "Content-Type: application/json" \ -d '{"amount": 10000}'
```

4. Withdraw QRIS ke Saldo Utama

Menarik saldo QRIS ke saldo utama akun OrderKuota.

POST `https://orkut.nickystore.my.id/api/orkut/withdraw`

Parameter	Tipe	Deskripsi
-----------	------	-----------

amount	number	Nominal withdraw (kelipatan Rp 1.000)
--------	--------	---------------------------------------

Auth	Cookie session (login dashboard)
------	----------------------------------

Request:

```
{ "amount": 50000 } Response:
{ "success": true, "data": { "results": { ... } } }
```

Python

```
import requests url = "https://orkut.nickystore.my.id/api/orkut/withdraw" headers
= { "Cookie": "token=TOKEN_DASHBOARD", "Content-Type": "application/json" } body =
{ "amount": 50000 } r = requests.post(url, json=body, headers=headers)
print(r.json())
```

JavaScript (fetch)

```
const res = await fetch("https://orkut.nickystore.my.id/api/orkut/withdraw", {
method: "POST", credentials: "include", headers: { "Content-Type":
"application/json" }, body: JSON.stringify({ amount: 50000 }) }); const data =
await res.json(); console.log(data);
```

cURL

```
curl -X POST "https://orkut.nickystore.my.id/api/orkut/withdraw" \ -H "Cookie:
token=TOKEN_DASHBOARD" \ -H "Content-Type: application/json" \ -d '{"amount":
50000}'
```

5. Cek Status Pembayaran

Memeriksa status payment berdasarkan ID. Auto-check: jika masih pending, sistem akan cek saldo QRIS dulu sebelum return.

```
POST https://orkut.nickystore.my.id/api/payment/check/:id
```

Parameter	Tipe	Deskripsi
:id	string (path)	ID payment yang didapat dari create payment

AuthCookie session atau header `x-api-key`

Response:

```
{ "success": true, "payment": { "id": "pay_abc123", "amount": 10000, "status": "succeed", "qr_image_url": "https://...", "balance_before": 500000, "balance_after": 510000 }, "increase": 10000 }
```

Python

```
import requests payment_id = "pay_abc123" url = f"https://orkut.nickystore.my.id/api/payment/check/{payment_id}" headers = { "x-api-key": "API_KEY_ANDA", "Content-Type": "application/json" } r = requests.post(url, headers=headers) payment = r.json()["payment"] print("Status:", payment["status"]) print("Nominal:", payment["amount"])
```

JavaScript (fetch)

```
const paymentId = "pay_abc123"; const res = await fetch(`https://orkut.nickystore.my.id/api/payment/check/${paymentId}`, { method: "POST", headers: { "x-api-key": "API_KEY_ANDA", "Content-Type": "application/json" } }); const data = await res.json(); console.log("Status:", data.payment.status);
```

cURL

```
curl -X POST "https://orkut.nickystore.my.id/api/payment/check/pay_abc123" \ -H "x-api-key: API_KEY_ANDA" \ -H "Content-Type: application/json"
```

Tips: Untuk integrasi bot, gunakan API Key via header `x-api-key`. Untuk dashboard, gunakan cookie session (login via form). Semua nominal dalam Rupiah (integer).