

# Snippe API Integration Guide

Mobile Money, Card, Dynamic QR, Payment Sessions & Payouts

API VERSION: 2026-01-25

## Kilichomo ndani ya PDF

- Muundo sahihi wa integration kutoka server yako kwenda Snippe.
- Flows za mobile money push, card redirect, QR scan-to-pay na hosted checkout.
- Payout/disbursement kwenda mobile money na bank accounts.
- Webhook verification, retries, idempotency, errors na security checklist.
- Mifano ya request/response na snippets za Node.js/PHP/Laravel-style.

**Prepared by Janet (ChatGPT)**

Date: 30 April 2026 | Language: Kiswahili + developer terms

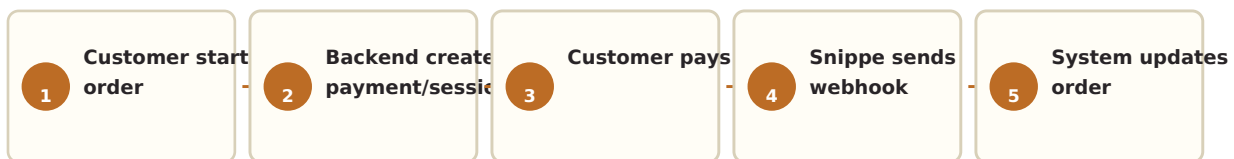
Source: docs.snippe.sh

## Muhtasari wa Integration

Jinsi ya kuunganisha Snippe API kwenye website, app au system ya biashara.

Snippe ni payment processing API inayokuwezesha kupokea malipo kupitia mobile money, card na dynamic QR, pamoja na kutuma disbursements/payouts kwenda mobile money na bank accounts. Base URL ya API ni <https://api.snippe.sh>, na requests zinahitaji API key kwenye header ya Authorization.

Kwenye implementation ya production, sehemu muhimu ni nne: create transaction, direct customer action (USSD push/redirect/QR), receive webhook, na reconcile status kwenye database yako. Usitumie redirect pekee kuthibitisha malipo; webhook ndiyo chanzo kikuu cha ukweli.



Njia mbili za integration:

- Payment Sessions / Hosted Checkout: njia rahisi na salama zaidi. Server yako inatengeneza session, inarudisha checkout\_url, mteja analipa kwenye UI ya Snippe.
- Direct Payments API: unajenga UI yako mwenyewe, halafu unatuma POST /v1/payments kwa mobile, card au dynamic-qr.
- Payouts / Disbursements: server yako inatuma pesa kwenda wallet au account baada ya balance na fees kuthibitishwa.

Mapendekezo ya haraka: kwa e-commerce au system ambayo unataka kuanza haraka tumia Payment Sessions; kwa product inayohitaji control kubwa ya UI tumia Direct Payments; kwa payroll, refunds au supplier payments tumia Payouts API.

## Yaliyomo

- 1. API base, authentication na scopes
- 2. Payment Sessions hosted checkout
- 3. Direct Payments: Mobile Money push, Card, Dynamic QR
- 4. Payouts: Mobile Money na Bank Transfer
- 5. Webhooks na signature verification
- 6. Idempotency, rate limits, errors na retry strategy
- 7. Database structure na implementation checklist
- 8. Code snippets za Node.js, PHP/Laravel na cURL

## 1. Base URL, Authentication na Scopes

API requests za Snippe zinaenda kwenye base URL hii:

```
https://api.snippe.sh
```

Kila request ya server-to-server inapaswa kuwa na API key kwenye Authorization header. API key isionekane kwenye frontend, browser, mobile app package, public GitHub repo au logs.

```
Authorization: Bearer snp_your_api_key_here  
Content-Type: application/json
```

Scopes muhimu wakati unatengeneza API key:

Scope	Matumizi
collection:read	Kusoma payments na balance
collection:create	Kutengeneza payment intents
disbursement:read	Kusoma payouts/disbursements
disbursement:create	Kutengeneza payouts

Tumia principle ya least privilege: system ya kupokea malipo pekee haihitaji disbursement:create. Kama utahitaji refunds/payouts, tengeneza key yenye scope husika na uihifadhi kwenye server environment variables.

### Mfano wa environment variables:

```
SNIPPE_API_KEY=snp_live_XXXXXXXXXXXXXXXXXX  
SNIPPE_WEBHOOK_SECRET=whsec_XXXXXXXXXXXXXXXXXX  
SNIPPE_WEBHOOK_URL=https://yourdomain.com/webhooks/snippe
```

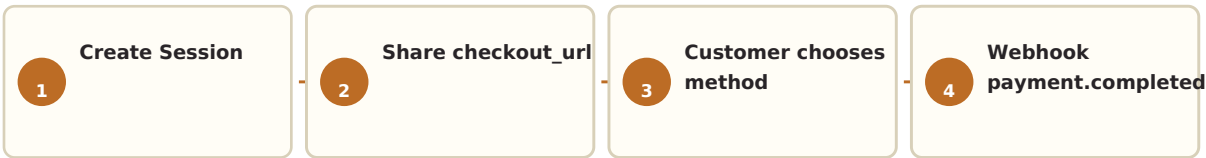
## Endpoints Overview

Kundi	Endpoint	Method	Maelezo
Payments	/v1/payments	POST	Create payment intent
Payments	/v1/payments/{reference}	GET	Get payment status
Payments	/v1/payments/balance	GET	Get account balance
Sessions	/api/v1/sessions	POST	Create hosted checkout session
Sessions	/api/v1/sessions/:reference	GET	Get session details
Payouts	/v1/payouts/send	POST	Create payout
Payouts	/v1/payouts/{reference}	GET	Get payout status
Payouts	/v1/payouts/fee?amount=...	GET	Calculate payout fee
Webhook	Your configured webhook_url	POST	Receive payment/payout events

## 2. Payment Sessions - Hosted Checkout

Njia rahisi: create session, redirect customer, receive webhook.

Payment Sessions zinakupa checkout page iliyojengwa tayari. Unatuma amount, allowed methods, customer info, redirect\_url na webhook\_url; Snippe inarudisha checkout\_url na payment\_link\_url.



### Create session request:

```
POST /api/v1/sessions
Authorization: Bearer <token> # or X-API-Key: <api_key>
Content-Type: application/json

{
  "amount": 50000,
  "currency": "TZS",
  "allowed_methods": ["mobile_money", "qr", "card"],
  "customer": {
    "name": "John Doe",
    "phone": "+255712345678",
    "email": "john@example.com"
  },
  "redirect_url": "https://yoursite.com/success",
  "webhook_url": "https://yoursite.com/webhooks/snippe",
  "description": "Order #12345",
  "metadata": { "order_id": "12345" },
  "expires_in": 3600
}
```

### Expected response:

```
{
  "code": 201,
  "data": {
    "reference": "sess_abc123def456",
    "status": "pending",
    "amount": 50000,
    "currency": "TZS",
    "checkout_url": "https://snippe.me/checkout/W0SzdUSHQm",
    "payment_link_url": "https://snippe.me/p/Ax7kM2",
    "allowed_methods": ["mobile_money", "qr", "card"],
    "metadata": { "order_id": "12345" }
  }
}
```

### Fields muhimu kwenye session:

Field	Required	Maelezo
amount	Yes/conditional	Kiasi kwa TZS integer. Required kwa fixed amount; optional kama allow_custom_amount ni true.

Field	Required	Maelezo
allowed_methods	No	mobile_money, qr, card. Default huweza kuwa mobile_money na qr.
webhook_url	Strongly recommended	Endpoint yako ya kupokea payment events.
redirect_url	Optional	Mteja kurudishwa baada ya checkout. Isitumike kama uthibitisho mkuu.
metadata	Recommended	Weka order_id, user_id, invoice_id kwa reconciliation.
billing	Card only	Inaweza kupunguza hatua za mteja kwenye card checkout.

Public checkout endpoints zinatumiwa na hosted checkout page: GET /checkout/:token, POST /checkout/:token/pay, GET /checkout/:token/status na cancel attempt. Hizi public endpoints hazihitaji authentication kwa sababu token ndiyo inatambulisha checkout.

## Payment Profiles

Payment Profiles zinatunza branding na default settings kama merchant\_name, logo, brand\_color, locale, allowed\_methods, redirect\_url na webhook\_url. Profile hutengenezwa kwenye dashboard, kisha unaweka profile\_id wakati wa create session. Branding fields hutoka kwenye profile, wakati config fields kama redirect\_url/webhook\_url zinaweza ku-overridden per request.

### 3. Direct Payments API

Mobile Money push, Card redirect na Dynamic QR.

Direct Payments API inafaa pale unapojenga checkout yako mwenyewe. Unatengeneza payment intent kupitia POST /v1/payments na kuamua aina ya payment\_type: mobile, card au dynamic-qr.

#### Payment status values:

Status	Maana
pending	Payment imetengenezwa na inasubiri hatua ya mteja.
completed	Malipo yamefanikiwa na fedha zimepokelewa.
failed	Malipo yameshindwa, yamekataliwa au yame-timeout.
voided	Payment imefutwa kabla haijakamilika.
expired	Payment ime-expire, kawaida baada ya saa 4.

#### 3.1 Mobile Money Push

Kwa mobile money, mteja anapokea USSD push kwenye simu yake, anaweka PIN, kisha Snippe inatuma webhook ya matokeo. Supported networks kwenye payments ni Airtel Money, M-Pesa, Mixx by Yas na Halotel.

```
POST /v1/payments
Authorization: Bearer <api_key>
Content-Type: application/json
Idempotency-Key: order-12345-pay1

{
  "payment_type": "mobile",
  "details": {
    "amount": 500,
    "currency": "TZS"
  },
  "phone_number": "255781000000",
  "customer": {
    "firstname": "FirstName",
    "lastname": "LastName",
    "email": "customer@email.com"
  },
  "webhook_url": "https://yoursite.com/webhooks/snippe",
  "metadata": { "order_id": "ORD-12345" }
}
```

Response itarudisha reference na status pending. Usi-mark order kama paid hadi webhook payment.completed ifike au GET status ithibitishe completed.

#### 3.2 Card Payments

Kwa card, unatuma billing/customer details, kisha response inarudisha payment\_url. Frontend yako inam-redirect customer kwenda payment\_url ili aingize card details kwenye secure checkout. Baada ya hapo Snippe inamtuma kwenye redirect\_url au cancel\_url, na webhook ndiyo inathibitisha mwisho.

```
POST /v1/payments
Authorization: Bearer <api_key>
Content-Type: application/json
Idempotency-Key: order-12345-card1

{
  "payment_type": "card",
  "details": {
```

```
"amount": 1000,
"currency": "TZS",
"redirect_url": "https://yourdomain.com/payment_done",
"cancel_url": "https://yourdomain.com/payment_failed"
},
"phone_number": "255781000000",
"customer": {
  "firstname": "FirstName",
  "lastname": "LastName",
  "email": "customer@email.com",
  "address": "Customer Address",
  "city": "Dar es Salaam",
  "state": "DSM",
  "postcode": "14101",
  "country": "TZ"
},
"webhook_url": "https://yoursite.com/webhooks/snippe",
"metadata": { "order_id": "ORD-12345" }
}
```

Card response fields muhimu: payment\_url kwa redirect, payment\_token kwa reference, payment\_qr\_code kama QR data, reference na expires\_at.

### 3.3 Dynamic QR

Dynamic QR ni nzuri kwa POS, maduka, counter, restaurant au kiosk. Response inarudisha payment\_qr\_code; render hiyo string kuwa QR image, mteja aiscan kwa mobile money app, kisha asubiri webhook.

```
POST /v1/payments
Authorization: Bearer <api_key>
Content-Type: application/json
Idempotency-Key: order-12345-qr1

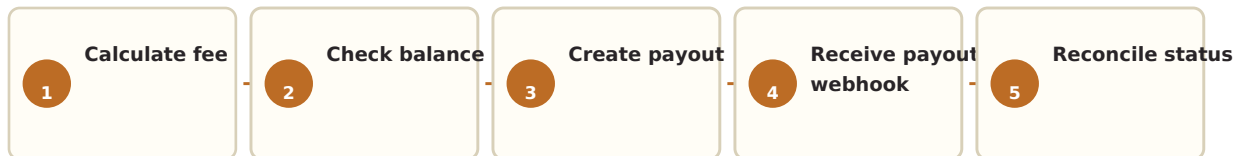
{
  "payment_type": "dynamic-qr",
  "details": {
    "amount": 500,
    "currency": "TZS",
    "redirect_url": "https://yourdomain.com/payment_done",
    "cancel_url": "https://yourdomain.com/payment_failed"
  },
  "customer": {
    "firstname": "FirstName",
    "lastname": "LastName",
    "email": "customer@email.com"
  },
  "webhook_url": "https://yoursite.com/webhooks/snippe",
  "metadata": { "order_id": "ORD-12345" }
}
```

Kwa QR, response inatoa payment\_qr\_code, payment\_url, payment\_token, reference na expires\_at. QR inaweza pia kuwa na hosted payment URL kama fallback.

## 4. Payouts / Disbursements

Kutuma pesa kwenda mobile money na bank accounts.

Disbursements API inatumika kutuma pesa kutoka kwenye balance yako kwenda mobile money wallet au bank account. Kabla ya ku-create payout, hakikisha una balance ya kutosha na calculate fee. Total inayokatwa ni amount + fees.



### Payout status values:

Status	Maana
pending	Payout imetengenezwa na inasubiri processing.
completed	Payout imefanikiwa na recipient amepokea fedha.
failed	Payout imeshindwa; angalia failure_reason kama ipo.
reversed	Payout ilifanikiwa lakini imereversed baadae.

### Calculate payout fee:

```
GET /v1/payouts/fee?amount=5000
Authorization: Bearer <api_key>
```

```
{
  "status": "success",
  "code": 200,
  "data": {
    "amount": 50000,
    "fee_amount": 1000,
    "total_amount": 51000,
    "currency": "TZS"
  }
}
```

### 4.1 Mobile Money Payout

Mobile payout inatumia channel mobile, recipient\_phone na recipient\_name. Docs zinaonesha networks za disbursement kama Airtel Money, Mixx by Yas na HaloPesa; provider inaweza kugunduliwa kulingana na phone number.

```
POST /v1/payouts/send
Authorization: Bearer <api_key>
Content-Type: application/json
Idempotency-Key: payout-emp-001-jan-2026
```

```
{
  "amount": 5000,
  "channel": "mobile",
  "recipient_phone": "255781000000",
}
```

```
"recipient_name": "Recipient Name",
"narration": "Salary payment January 2026",
"webhook_url": "https://yoursite.com/webhooks/snippe",
"metadata": {
  "employee_id": "EMP-001",
  "payroll_id": "PAY-2026-01"
}
```

Response itarudisha reference, external\_reference, status pending, amount, fees, total na channel.provider. Hifadhi reference kwenye database yako kwa status polling/reconciliation.

## 4.2 Bank Transfer Payout

Bank transfer payout hutuma pesa moja kwa moja kwenye account ya benki. Unahitaji channel bank, recipient\_bank code, recipient\_account na recipient\_name. Docs zinaorodhesha zaidi ya benki 40 za Tanzania kama CRDB, NMB, NBC, ABSA na nyingine.

```
POST /v1/payouts/send
Authorization: Bearer <api_key>
Content-Type: application/json
Idempotency-Key: inv-2026-001-pay

{
  "amount": 5000,
  "channel": "bank",
  "recipient_bank": "ABSA",
  "recipient_account": "0200000000",
  "recipient_name": "Recipient Name",
  "narration": "Invoice payment INV-2026-001",
  "webhook_url": "https://yoursite.com/webhooks/snippe",
  "metadata": { "invoice_id": "INV-2026-001" }
}
```

## 5. Webhooks

Uthibitisho wa kweli wa payment na payout events.

Webhook ni HTTP POST ambayo Snippe inatuma kwenye `webhook_url` yako tukio likitokea. Server yako inapaswa kurudisha 2xx status ndani ya muda mfupi. Kama endpoint yako itashindwa au ita-timeout, Snippe itafanya retry kwa exponential backoff hadi attempts 5.

Event	Maelezo
payment.completed	Payment imefanikiwa. Mark order/invoice as paid.
payment.failed	Payment imeshindwa. Ruhusu retry au notify customer.
payment.voided	Payment imefutwa kabla haijakamilika.
payment.expired	Payment ime-expire. Tengeneza payment mpya kama mteja anataka kurudia.
payout.completed	Payout imefanikiwa. Mark disbursement as completed.
payout.failed	Payout imeshindwa; funds zinaweza kurudi kwenye balance.
payout.reversed	Payout iliyokamilika imereversed. Fanya reconciliation.

### Headers muhimu za webhook:

Header	Matumizi
X-Webhook-Event	Event type mfano payment.completed
X-Webhook-Timestamp	Unix timestamp ya tukio; tumia kuzuia replay attacks
X-Webhook-Signature	HMAC-SHA256 signature ya kuthibitisha kuwa request imetoka Snippe
Content-Type	application/json

### Payload current format:

```
{
  "id": "evt_alb2c3d4e5f6g7h8i9j0",
  "type": "payment.completed",
  "api_version": "2026-01-25",
  "created_at": "2026-01-24T10:30:00Z",
  "data": {
    "reference": "pi_alb2c3d4e5f6",
    "external_reference": "SEL123456789",
    "status": "completed",
    "amount": { "value": 50000, "currency": "TZS" },
    "settlement": {
      "gross": { "value": 50000, "currency": "TZS" },
      "fees": { "value": 1000, "currency": "TZS" },
      "net": { "value": 49000, "currency": "TZS" }
    },
    "channel": { "type": "mobile_money", "provider": "mpesa" },
    "customer": { "phone": "+255712345678", "name": "John Doe" },
    "metadata": { "order_id": "ORD-12345" },
    "completed_at": "2026-01-24T10:30:00Z"
  }
}
```

Important: kwenye request ya create payment, amount ni integer tu; kwenye webhook payload, data.amount ni object yenye value na currency. Kwa hiyo code yako itumie `event.data.amount.value` na `event.data.amount.currency`.

### Signature verification logic

- Soma raw request body kama string/Buffer. Usiparse na ku-serialize upya kabla ya signature check.
- Tengeneza message kwa format: {timestamp}.{raw\_body}.
- Compute HMAC-SHA256 kwa webhook signing key yako.
- Compare computed signature na X-Webhook-Signature kwa constant-time comparison.
- Reject timestamp ya zaidi ya dakika 5 ili kuzuia replay attacks.

```
// Node.js / Express raw-body example
app.post('/webhooks/snippe', express.raw({ type: 'application/json' })), (req, res) => {
  const rawBody = req.body.toString();
  const timestamp = req.header('X-Webhook-Timestamp');
  const signature = req.header('X-Webhook-Signature');

  // verifyWebhook(rawBody, timestamp, signature, process.env.SNIPPE_WEBHOOK_SECRET)
  // then process event.type: payment.completed, payment.failed, payout.completed, etc.

  res.status(200).send('OK');
};
```

Webhook handling best practice: respond 200 quickly, kisha process asynchronously. Hifadhi event.id/reference ili ku-deduplicate kwa sababu event moja inaweza kutumwa zaidi ya mara moja.

## 6. Idempotency, Rate Limits na Error Handling

Idempotency-Key ni muhimu kwa requests za kutengeneza payment au payout. Ukiretry kwa network error, key ile ile + body ile ile itarudisha response ile ile badala ya kutengeneza transaction duplicate. Kwa payouts, key ni valid kwa saa 24. Docs pia zinaonya kuwa Idempotency-Key ndefu zaidi ya characters 30 inaweza kurudisha PAY\_001.

```
POST /v1/payments
Idempotency-Key: order123pay1

POST /v1/payouts/send
Idempotency-Key: payemp001jan26
```

Rate limit ya API ni 60 requests per minute. Hifadhi na respect headers kama X-Ratelimit-Limit, X-Ratelimit-Remaining na X-Ratelimit-Reset; kwa 429 tumia exponential backoff.

### Common errors:

HTTP	error_code	Maana / Fix
400	validation_error	Request ina field missing/invalid. Validate amount, phone, URLs na required fields.
401	unauthorized	API key missing, invalid au expired. Hakikisha Authorization header ipo.
403	insufficient_scope	API key haina scope inayohitajika, mfano disbursement:create.
404	not_found	Payment, payout au session haijapatikana. Angalia reference.
409	conflict	State conflict, mfano payment already in progress.
422	validation_error	Idempotency key imetumika na body tofauti. Tumia key mpya kwa request mpya.
429	rate_limit_exceeded	Too many requests. Backoff na tumia X-Ratelimit-Reset.
500	payment_failed / PAY_001	Processor issue, insufficient balance, au idempotency key ndefu sana. Retry kwa backoff baada ya validation.

### Validation rules muhimu:

- Minimum amount: payments 500 TZS; payouts 5,000 TZS.
- Currency inayoungwa mkono kwenye docs ni TZS; amount iwe integer kwenye smallest currency unit.
- Phone number iwe Tanzanian format: +255XXXXXXXXX au 255XXXXXXXXX.
- Production webhook\_url, redirect\_url na cancel\_url ziwe HTTPS na zisizidi characters 500.

## 7. Database Structure na Backend Logic

System yako inahitaji kutunza transactions locally ili uweze kufanya reconciliation hata kama webhook imechelewa, customer amefunga browser, au provider ame-delay status. Huu ni mfano wa tables za kuanzia.

### payments table

Column	Type	Purpose
id	uuid/bigint	Primary key ya system yako
order_id	string/uuid	Order/invoice inayolipiwa
snippe_reference	string unique	Reference kutoka Snippe
payment_type	string	mobile, card, dynamic-qr au session
amount	integer	Amount in TZS
currency	string	TZS
status	string	pending, completed, failed, expired, voided
customer_phone	string	Phone ya mteja
metadata	json	order_id, user_id, custom tracking
raw_response	json	Response ya create payment/session
completed_at	timestamp nullable	Wakati payment imekamilika

### payouts table

Column	Type	Purpose
id	uuid/bigint	Primary key
snippe_reference	string unique	Reference ya payout
channel	string	mobile au bank
amount	integer	Amount ya recipient
fees	integer	Transaction fees
total	integer	Amount + fees
recipient_phone/account	string	Wallet number au bank account
recipient_name	string	Jina la mpokeaji
status	string	pending, completed, failed, reversed
metadata	json	employee_id, invoice_id, payroll_id
raw_response	json	Response kutoka Snippe

### webhook\_events table

Column	Type	Purpose
event_id	string unique	Deduplication ya webhook event
event_type	string	payment.completed, payout.failed, n.k.

Column	Type	Purpose
reference	string	Payment/payout/session reference
payload	json	Full webhook payload
processed_at	timestamp nullable	Imeprocesssiwa lini
processing_status	string	received, processed, failed

## Backend sequence ya payment completed:

- Receive webhook na verify signature.
- Check kama event\_id tayari ipo; kama ipo, return 200 bila ku-process tena.
- Find payment kwa data.reference au metadata.order\_id/session\_reference.
- Update payment status = completed, amount, settlement fees/net, completed\_at.
- Mark order/invoice as paid ndani ya database transaction moja.
- Send receipt/notification kwa customer kama inahitajika.

## 8. Implementation Snippets

### Node.js SDK

```
import { Snippe } from '@snippe/sdk';

const snippe = new Snippe({
  apiKey: process.env.SNIPPE_API_KEY,
  webhookUrl: 'https://yourdomain.com/webhooks/snippe',
});

const payment = await snippe.payments.create({
  payment_type: 'mobile',
  details: { amount: 500, currency: 'TZS' },
  phone_number: '255781000000',
  customer: {
    firstname: 'Jane',
    lastname: 'Doe',
    email: 'jane@example.com',
  },
  metadata: { order_id: 'ORD-12345' },
}, { idempotencyKey: 'ord12345pay1' });
```

### Node.js payout:

```
const payout = await snippe.payouts.send({
  amount: 50000,
  channel: 'mobile',
  recipient_phone: '255781000000',
  recipient_name: 'Employee Name',
  narration: 'Salary January 2026',
});

const { fee_amount, total_amount } = await snippe.payouts.fee({ amount: 50000 });
```

### PHP SDK

```
composer require snippe/snippe-php

use Snippe\Snippe;

$snippe = new Snippe(getenv('SNIPPE_API_KEY'));

$payment = $snippe->mobileMoney(5000, '0754123456')
->customer('John Doe', 'john@email.com')
->idempotencyKey('ord123pay1')
->send();
```

### PHP card redirect:

```
$payment = $snippe->card(10000)
->phone('0754123456')
->customer('John Doe', 'john@email.com')
->billing('123 Main Street', 'Dar es Salaam', 'DSM', '14101', 'TZ')
->redirectTo('https://yoursite.com/success', 'https://yoursite.com/cancel')
->send();

header('Location: ' . $payment->paymentUrl());
exit;
```

### Laravel-style service skeleton

```
class SnippePaymentService
{
    public function createMobilePayment(Order $order): array
    {
        $response = Http::withToken(config('services.snippe.key'))
        ->withHeaders([
```

```
'Idempotency-Key' => 'ord' . $order->id . 'pay1',
])
->post('https://api.snippe.sh/v1/payments', [
'payment_type' => 'mobile',
'details' => [
'amount' => (int) $order->amount,
'currency' => 'TZS',
],
'phone_number' => $order->customer_phone,
'customer' => [
'firstname' => $order->customer_first_name,
'lastname' => $order->customer_last_name,
'email' => $order->customer_email,
],
'webhook_url' => route('webhooks.snippe'),
'metadata' => ['order_id' => (string) $order->id],
])->throw()->json();

return $response['data'];
}
}
```

## 9. Testing na Go-Live Checklist

### Testing checklist

- Test authentication: invalid key lazima irudishe 401; key bila scope irudishe 403.
- Test mobile payment: create intent, hakikisha status ni pending, kisha webhook iweze ku-update order.
- Test card: redirect payment\_url, cancel\_url na redirect\_url zifanye kazi bila ku-mark paid kabla ya webhook.
- Test QR: render payment\_qr\_code kuwa QR image; customer scan flow ifanye kazi.
- Test payout fee: calculate fee kabla ya payout na verify total amount.
- Test insufficient balance kwa payout na user-friendly message.
- Test webhook signature: invalid signature irudishe 400; valid signature irudishe 200.
- Test duplicate webhook: event ile ile ikija mara mbili isi-update order mara mbili.
- Test idempotency: retry request na same key/body isilete duplicate transaction.
- Test 429/5xx retry logic kwa exponential backoff.

### Production checklist

- API keys ziko kwenye environment variables tu, siyo frontend wala GitHub.
- Webhook URL ni HTTPS na ina raw-body support kwa signature verification.
- Database ina unique indexes kwa snippe\_reference na webhook event\_id.
- Logs hazihifadhi API key wala webhook secret.
- Una dashboard/command ya reconcile payments/payouts kwa GET status.
- Una monitoring ya 4xx, 5xx, webhook failures na abandoned webhook attempts.
- Idempotency keys ni  $\leq 30$  characters na zina strategy consistent.

### Recommended production workflow

Scenario	Recommended Approach
Online store / SaaS subscription	Payment Sessions + metadata.order_id + webhook
Custom checkout UI	Direct Payments API + webhook + polling fallback
POS / shop counter	Dynamic QR + webhook + status screen
Salary/refund/supplier payments	Payout fee -> balance check -> payout -> webhook
Laravel system	Service class + queued webhook processing + DB transactions

## 10. Sources

PDF hii imeandaliwa kwa kusoma Snippe Documentation version 2026-01-25 na kurahisisha maelezo kwa implementation ya developer. Taarifa za msingi zilizotumika zinatoka kwenye kurasa zifuatazo:

Topic	Source URL
Main docs / base URL / auth overview	<a href="https://docs.snippe.sh/docs/2026-01-25">https://docs.snippe.sh/docs/2026-01-25</a>
Authentication	<a href="https://docs.snippe.sh/docs/2026-01-25/authentication">https://docs.snippe.sh/docs/2026-01-25/authentication</a>
Payment Sessions	<a href="https://docs.snippe.sh/docs/2026-01-25/sessions">https://docs.snippe.sh/docs/2026-01-25/sessions</a>
Payment Profiles	<a href="https://docs.snippe.sh/docs/2026-01-25/sessions/profiles">https://docs.snippe.sh/docs/2026-01-25/sessions/profiles</a>
Payments Overview	<a href="https://docs.snippe.sh/docs/2026-01-25/payments">https://docs.snippe.sh/docs/2026-01-25/payments</a>
Mobile Money Payments	<a href="https://docs.snippe.sh/docs/2026-01-25/payments/mobile-money">https://docs.snippe.sh/docs/2026-01-25/payments/mobile-money</a>
Card Payments	<a href="https://docs.snippe.sh/docs/2026-01-25/payments/card">https://docs.snippe.sh/docs/2026-01-25/payments/card</a>
Dynamic QR	<a href="https://docs.snippe.sh/docs/2026-01-25/payments/dynamic-qr">https://docs.snippe.sh/docs/2026-01-25/payments/dynamic-qr</a>
Disbursements	<a href="https://docs.snippe.sh/docs/2026-01-25/disbursements">https://docs.snippe.sh/docs/2026-01-25/disbursements</a>
Mobile Money Disbursements	<a href="https://docs.snippe.sh/docs/2026-01-25/disbursements/mobile-money">https://docs.snippe.sh/docs/2026-01-25/disbursements/mobile-money</a>
Bank Transfer Disbursements	<a href="https://docs.snippe.sh/docs/2026-01-25/disbursements/bank-transfer">https://docs.snippe.sh/docs/2026-01-25/disbursements/bank-transfer</a>
Webhooks	<a href="https://docs.snippe.sh/docs/2026-01-25/webhooks">https://docs.snippe.sh/docs/2026-01-25/webhooks</a>
SDKs	<a href="https://docs.snippe.sh/docs/2026-01-25/sdks">https://docs.snippe.sh/docs/2026-01-25/sdks</a>
Error Handling	<a href="https://docs.snippe.sh/docs/2026-01-25/error-handling">https://docs.snippe.sh/docs/2026-01-25/error-handling</a>

Note: Kabla ya production deployment, hakikisha account yako ya Snippe imehakikiwa, API key scopes ni sahihi, webhook secret ipo, na business flow yako imejaribiwa kwa amounts ndogo.