ZayanPay

AUTO PAYMENT VERIFICATION SYSTEM

Developer API & Integration Reference
Version: **2.4.0 (Latest Release)**
Platform: **Bangladesh Multi-MFS Auto Gateway**
• PRODUCTION READY

BASE URL	CHECKOUT ENDPOINT	VERIFY ENDPOINT	AUTH HEADER
`http://127.0.0.1:8088`	`POST /api/checkout/redirect`	`POST /api/verify-payment`	`ZAYANPAY-API-KEY`

1. Overview & Authentication

ZayanPay provides an automated payment verification infrastructure for Bangladeshi online merchants. It connects directly to personal and merchant bKash, Nagad, Rocket, Upay, and Cellfin accounts through an Android companion app, verifying incoming customer payments instantly (0.1s) with **0% gateway transaction fees**.

All HTTP requests sent to the ZayanPay REST API must include the following headers:

Header Name	Example Value	Description
`Content-Type`	`application/json`	Must be application/json for all POST requests.
`Accept`	`application/json`	Specifies that the client expects JSON responses.
`ZAYANPAY-API-KEY`	`zp_live_xxxxxxxxxxxx`	Your private merchant API key obtained from Merchant Portal → Developer Keys.

2. Checkout Redirect API (Create Invoice)

Use this endpoint to generate a hosted checkout payment session. Redirect your customer to the returned `pp_url` where they choose their preferred wallet (bKash, Nagad, Rocket, etc.) and complete the payment.

POST `http://127.0.0.1:8088/api/checkout/redirect`

Request Parameters

Field	Type	Required	Description
`full_name`	string	**Yes**	Customer full name (e.g. Tanvir Ahmed).
`email_address`	string	**Yes**	Customer email address.
`mobile_number`	string	**Yes**	Customer mobile phone number (e.g. 01712345678).
`amount`	numeric	**Yes**	Payment amount in BDT (e.g. 500 or 1200.00).
`currency`	string	No	Default currency code is `BDT`.
`return_url`	string (URL)	**Yes**	Customer redirect destination upon payment completion.
`webhook_url`	string (URL)	No	Asynchronous server-to-server IPN listener on your website.
`metadata`	object	No	Arbitrary key-value pairs (e.g. `{"order_id": "ORD-1001"}`).

Success Response (200 OK)

```
{
  "pp_id": "291583772135977897208017178",
  "pp_url": "http://127.0.0.1:8088/checkout/291583772135977897208017178",
  "status": "pending",
  "expires_at": "2026-09-16T18:30:00Z"
}
```

3. Payment Verification API

Always verify transaction authenticity directly from your backend before processing orders or activating services.

POST `http://127.0.0.1:8088/api/verify-payment`

Request Parameters

Field	Type	Required	Description
`pp_id`	string	Yes	Unique payment session ID received in redirect query string or checkout response.

Success Response (200 OK)

```
{
  "status": "completed",
  "pp_id": "291583772135977897208017178",
  "trx_id": "9J8A7K210L",
  "payment_method": "bkash",
  "amount": "500.00",
  "currency": "BDT",
  "customer_mobile": "01712345678"
}
```

4. Step-by-Step Laravel Full Integration Guide

Follow these 5 simple steps to integrate ZayanPay into any Laravel 9, 10, 11, or 12 application:

1 Environment Configuration (.env)

```
ZAYANPAY_BASE_URL=http://127.0.0.1:8088
ZAYANPAY_API_KEY=zp_live_your_merchant_api_key
ZAYANPAY_CURRENCY=BDT
```

2 Service Class (app/Services/ZayanPayService.php)

```
<?php
namespace App\Services;

use Illuminate\Support\Facades\Http;
use RuntimeException;

class ZayanPayService
{
    protected string $baseUrl;
    protected string $apiKey;

    public function __construct()
    {
        $this->baseUrl = rtrim(env('ZAYANPAY_BASE_URL', 'http://127.0.0.1:8088'), '/');
        $this->apiKey  = (string) env('ZAYANPAY_API_KEY');
    }

    public function createPayment(string $name, string $email, string $phone, float $amount, string $returnUr
l, ?string $webhookUrl = null, array $metadata = []): array
    {
```

```
        $response = Http::timeout(15)
            ->withHeaders([
                'ZAYANPAY-API-KEY' => $this->apiKey,
                'Accept' => 'application/json',
            ])
            ->post("{$this->baseUrl}/api/checkout/redirect", [
                'full_name'     => $name,
                'email_address' => $email,
                'mobile_number' => $phone,
                'amount'        => (string)$amount,
                'currency'      => env('ZAYANPAY_CURRENCY', 'BDT'),
                'return_url'    => $returnUrl,
                'webhook_url'   => $webhookUrl ?? '--',
                'metadata'      => $metadata,
            ]);

        if ($response->failed()) {
            throw new RuntimeException($response->json('error.message') ?? 'Failed to initiate ZayanPay check
out.');
        }

        return $response->json();
    }

    public function verifyPayment(string $ppId): array
    {
        $response = Http::timeout(10)
            ->withHeaders([
                'ZAYANPAY-API-KEY' => $this->apiKey,
                'Accept' => 'application/json',
            ])
            ->post("{$this->baseUrl}/api/verify-payment", [
                'pp_id' => $ppId,
            ]);

        return $response->json();
    }
}
```

3 Controller (app/Http/Controllers/PaymentController.php)

```php
<?php
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Services\ZayanPayService;

class PaymentController extends Controller
{
    protected ZayanPayService $zayanPay;

    public function __construct(ZayanPayService $zayanPay)
    {
        $this->zayanPay = $zayanPay;
    }

    // 1. Submit order & redirect to ZayanPay
    public function pay(Request $request)
    {
        $orderId = 'ORD-' . time();
        $amount  = (float) $request->input('amount', 500);

        $checkout = $this->zayanPay->createPayment(
            name: $request->input('name', 'Tanvir Ahmed'),
            email: $request->input('email', 'customer@example.com'),
            phone: $request->input('phone', '01712345678'),
            amount: $amount,
            returnUrl: route('payment.callback'),
            webhookUrl: url('/api/payment/webhook'),
            metadata: ['order_id' => $orderId]
        );

        session(['pending_order' => ['order_id' => $orderId, 'pp_id' => $checkout['pp_id']]]);
        return redirect()->away($checkout['pp_url']);
    }

    // 2. Customer Return Callback
    public function callback(Request $request)
    {
        $trxRef = $request->query('transaction_ref');
        $status = $request->query('pp_status');

        $verification = $this->zayanPay->verifyPayment($trxRef);

        if (($verification['status'] ?? '') === 'completed' || $status === 'completed') {
            $trxId = $verification['trx_id'] ?? $trxRef;
            // Mark order paid in database
            return view('payment.success', compact('trxId'));
        }

        return redirect('/')->with('error', 'Payment failed or cancelled.');
    }

    // 3. Server-to-Server IPN Webhook
    public function webhook(Request $request)
    {
        $payload = $request->all();
        if (($payload['status'] ?? '') === 'completed') {
            $orderId = $payload['metadata']['order_id'] ?? null;
            // Fulfill order
            return response()->json(['status' => 'success']);
```

```
        }
        return response()->json(['status' => 'received']);
    }
}
```

4 Register Routes (routes/web.php)

```
use App\Http\Controllers\PaymentController;

Route::post('/payment/checkout', [PaymentController::class, 'pay'])->name('payment.checkout');
Route::get('/payment/callback', [PaymentController::class, 'callback'])->name('payment.callback');
Route::post('/api/payment/webhook', [PaymentController::class, 'webhook'])->name('payment.webhook');
```

5 Exclude Webhook Route from CSRF Verification

In Laravel 11 / 12 (`bootstrap/app.php`):

```
->withMiddleware(function (Middleware $middleware): void {
    $middleware->validateCsrfTokens(except: [
        'api/payment/webhook', // Exclude ZayanPay IPN
    ]);
})
```