EVzone Pay – PayKit SDK (Wallet Checkout)

This SDK lets your app accept payments from customers using their **EVzone Pay** wallet. It wraps a 3-step server flow: **Session** → **Quote** → **Charge**, and ships a ready UI (`<WalletPaymentForm />`).

---

1) For Developers (Technical)

Quick Start

```
import { WalletPaymentForm } from 'evzone-pay-ss9';

<WalletPaymentForm
  publishableKey="pk_test_123"
  enterpriseWalletNo="EVZ-123456"
  userWalletId="user_789"
  amount={60000}
  type="Purchase"
  particulars="SDK smoke test"
  merchantLogo="https://…/logo.png"
  onClose={() => setOpen(false)}
  onSuccess={(payload) => console.log('SUCCESS', payload)}
/>
```

SDK config: set your API base once:

```
// src/sdk/constants.js (in your consumer app after install)
export const API_BASE_URL = 'http://localhost:4000'; // prod: https://api.evzone-pay.com
export const DEFAULT_TIMEOUT_MS = 12000;
```

> **Server mount required** The API must expose routes under **`/api/v1/paykit`** (see endpoints below). Ensure CORS allows header **`Idempotency-Key`** on `/sdk/tx/charge`.

Components & Client

- `WalletPaymentForm` – drop-in checkout modal that orchestrates the flow.
- `paykitClient` – low-level client with three methods:
  - `initSession({ enterpriseWalletNo, userWalletId })`
  - `quote({ sessionId, amount })`
  - `charge({ sessionId, quoteId, passcode, idempotencyKey? })`

Required Inputs from Your Platform

Field	Who provides	Example	Notes
`publishableKey`	You	`pk_test_123`	issued to the merchant/integration
`enterpriseWalletNo`	You	`EVZ-123456`	the merchant/enterprise wallet to receive funds
`userWalletId`	You/Customer	`user_789`	the **paying** customer's EVzone wallet id
`amount`	You	`60000`	billed amount in the **billing currency** (server defaults to user's currency)
`type`	Optional	`Purchase`	shown in UI/receipt
`particulars`	Optional	`SDK smoke test`	shown in UI/receipt
`merchantLogo`	Optional	`https://…`	UI only

Endpoints (your route)

All routes are mounted below `/api/v1/paykit`:

1. **POST `/sdk/session/init`** Body:

```
{
  "publishableKey": "pk_test_123",
  "enterpriseWalletNo": "EVZ-123456",
  "userWalletId": "user_789",
  "billingCurrency": "UGX"  // optional; defaults to user's currency
}
```

Returns:

```
{
  "ok": true,
  "sessionId": "sess_abc",
  "enterprise": { "walletNo":"EVZ-123456", "name":"Xtraordinary Ltd",
"currency":"UGX" },
  "user": { "walletId":"user_789", "name":"Jane Smith",
"email":"jane@x.com", "balance":50000, "currency":"UGX" },
  "billingCurrency": "UGX",
  "rates": { "charges": { "taxPct": 0.025, "walletFeePct": 0.015 } },
  "expiresAt": "2025-09-12T12:34:56.000Z"
}
```

2. **POST `/sdk/tx/quote`** Body:

```
{ "sessionId": "sess_abc", "amount": 60000 }
```

Returns (abridged):

```
{
  "ok": true,
  "quoteId": "qt_123",
  "billingCurrency":"UGX",
  "userCurrency":"UGX",
  "enterpriseCurrency":"UGX",
  "fx": { "userToBilling": 1, "billingToEnterprise": 1,
"lockedForSec": 300 },
  "breakdown": {
    "subtotal": 60000,
    "taxPct": 0.025,
    "tax": 1500,
    "walletFeePct": 0.015,
    "walletFee": 900,
    "total": 60000
  },
  "expiresAt":"2025-09-12T12:39:56.000Z"
}
```

> **Note:** fees/tax are *informational* in this demo; **payable total = subtotal**.

3. **POST** `/sdk/tx/charge` Headers: `Idempotency-Key: idmp_169...` *(recommended)* Body:

```
{ "sessionId":"sess_abc", "quoteId":"qt_123", "passcode":"123456" }
```

Returns:

```
{
  "ok": true,
  "receipt": {
    "transactionId": "W-123456789",
    "timestamp": "2025-09-12T12:35:10.000Z",
    "billing": { "amount": 60000, "currency":"UGX" },
    "user": { "walletId":"user_789", "debited":60000,
"currency":"UGX", "newBalance":- },
    "enterprise": { "walletNo":"EVZ-123456", "credited":60000,
"currency":"UGX" },
    "fx": { "userToBilling":1, "billingToEnterprise":1 },
    "fees": { "tax":1500, "walletFee":900 }
  }
}
```

Data Flow (what happens)

1. **Open form** → `WalletPaymentForm` shows a loading overlay and calls **`/sdk/session/init`**.
2. **Summary** → User sees merchant + amount. On **Confirm**, the SDK calls **`/sdk/tx/quote`**.
3. **Balance check** → SDK compares `user.balance` vs. `quote.total`:
   - If insufficient: **Insufficient Funds** modal.
   - Else: shows **passcode** entry.
4. **Charge** → User enters passcode; SDK calls **`/sdk/tx/charge`** with an **Idempotency-Key**.
5. **Result** →
   - OK → **Payment Successful** modal (+ `onSuccess(receipt)` callback).
   - 402 → **Insufficient Funds** modal.
   - 403 → **Payment Failed** (*invalid passcode*).
   - Other errors → **Payment Failed**.

Expiry: session = 15 min, quote = 5 min. If expired, re-init/re-quote.

Status & Error Codes

Code / HTTP	Meaning	Typical UI
`200 {ok:true}`	Success	Success modal
`401 SESSION_EXPIRED`	Session missing/expired	Re-open checkout
`400 INVALID_AMOUNT`	Amount ≤ 0	Fix amount and retry
`400/404 FX_UNAVAILABLE / ENTERPRISE_WALLET_NOT_FOUND / USER_NOT_FOUND`	Config issue	Show error
`402 INSUFFICIENT_FUNDS`	Not enough balance	Insufficient Funds modal
`403 INVALID_PASSCODE`	Wrong passcode	Failed modal
`5xx INTERNAL_ERROR`	Server issue	Failed modal

CORS note: allow header **`Idempotency-Key`** in preflight for `/sdk/tx/charge`.

Security Notes

- **Passcode** is sent only to `/sdk/tx/charge`; the SDK doesn't store it.
- Use **HTTPS** in production.

- Use **Idempotency-Key** to protect against duplicate submits/retries.
- Treat `publishableKey` as public; server still validates it per merchant.

---

2) For Normal Users (Non-Technical)

What is happening when you pay with EVzone Pay?

1. On the checkout, choose **Pay with EVzone Wallet**.
2. You'll see a summary with the business name and the amount to pay.
3. Tap **Confirm**, then enter your **EVzone wallet passcode**.
4. If you have enough balance, the payment goes through and you'll see **Payment Successful**.
5. If your balance is too low, you'll see **Insufficient Funds** with options to add money or try again.
6. If the passcode is wrong, you'll see **Payment Failed** and can try again.

What do you need?

- An EVzone Pay wallet and your passcode.
- Enough money in the wallet for the purchase.

What do you receive?

- A clear confirmation on screen.
- The business gets the payment instantly, and your wallet balance updates.

FAQ (Short)

- **Can I retry if internet drops?** Yes. The system prevents duplicate charges automatically.

- **Where does my money go?** From your EVzone wallet to the business's EVzone enterprise wallet.

- **Why did it fail?** Usually because of a wrong passcode, not enough balance, or an expired session (just reopen and try again).
