EVzone PayKit — Web (npm SDK)

A drop-in React component that lets your customers pay with their **EVzone Wallet**. It renders a complete, branded flow (Summary → Passcode → Processing → Success/Failed/Insufficient Funds) with validations and UI transitions.

Requirements

- React 17+ (or 18+)

- Provide these values from **your platform** for the current payer:

  - `publishableKey`
  - `enterpriseWalletNo` (wallet **receiving** the money)
  - `userWalletId` (wallet **sending** the money)
  - `amount` (total to charge)
  - `type` (e.g., `"Purchase"`, `"Subscription"`)
  - `particulars` (what's being purchased)
  - `merchantLogo` (URL to your company/brand logo)

> **Note:** Do **not** provide currency. It is determined by EVzone based on the wallet IDs you pass.

Installation

```
npm install evzone-pay-ss9
```

Quick start

Render the form when the user clicks **Pay Now**.

```jsx
// src/Home.jsx
import React, { useMemo, useState } from 'react';
import { WalletPaymentForm } from 'evzone-pay13';

const PUBLISHABLE_KEY = 'pk_test_123';
const ENTERPRISE_WALLET_NO = 'EVZ-123456';
const USER_WALLET_ID = 'user_789';

export default function Home() {
  const [show, setShow] = useState(false);
  const amount = 60000;

  const txn = useMemo(
```

```
  () => ({
      amount,
      type: 'Purchase',
      particulars: '2 Computers',
      merchantLogo:

'https://res.cloudinary.com/dlfa42ans/image/upload/v1753798918/boys_zmnhiz.
jpg',
    }),
    [amount]
  );

  return (
    <div style={{ padding: 24 }}>
      <h1>EVzone SDK for Commercial or Online Shopping</h1>
      <p>Amount: UGX {amount.toLocaleString()}</p>

      <button onClick={() => setShow(true)}>Pay Now</button>

      {show && (
        <WalletPaymentForm
          // REQUIRED identifiers (drive session init)
          publishableKey={PUBLISHABLE_KEY}
          enterpriseWalletNo={ENTERPRISE_WALLET_NO}
          userWalletId={USER_WALLET_ID}

          // Transaction inputs
          amount={txn.amount}
          type={txn.type}
          particulars={txn.particulars}
          merchantLogo={txn.merchantLogo}

          // UX handlers
          onClose={() => setShow(false)}
          onSuccess={(payload) => {
            console.log('SUCCESS payload:', payload);

          }}
        />
      )}
    </div>
  );
}
```

What you must provide (from your app)

- **publishableKey** – Your EVzone publishable key.
- **enterpriseWalletNo** – The enterprise wallet **receiving** the payment.
- **userWalletId** – The customer's wallet **sending** the payment.
- **amount** – The total amount to charge (sum of items/services).

- `type` – Transaction type (e.g., `"Purchase"`, `"Subscription"`).
- `particulars` – A human-readable description of what's being purchased.
- `merchantLogo` – URL to your brand's logo (displayed in the modals).

> **Currency:** Do **not** pass a currency prop. The EVzone backend derives the correct currency from the wallet IDs you provide.

License & Support

- **License:** MIT
- **Support:** support@evzoneafrica.com