

Here's a clean, integration-only README that matches your latest SDK design (no server flow details, no `userWalletId` on the host app, currency auto-detected, and optional support contact shown in the "Cannot Continue" modal).

EVzone PayKit — Web (React 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 built-in validations and smooth UI.

Requirements

- React 17+ (or 18+)
- From **your app**, provide:
 - `publishableKey`
 - `enterpriseWalletNo` (the enterprise wallet **receiving** funds)
 - `amount` (the total to charge)
 - `type` (e.g. "Purchase", "Subscription")
 - `particulars` (what the customer is paying for)
 - `merchantLogo` (URL to your logo)
 - *(optional)* `supportEmail`, `supportPhone` (shown if the checkout can't start)

Do not pass a currency. EVzone determines the currency from the wallets.

Do not pass a user wallet ID or user number. The SDK handles the user internally (via a signed-in check and a secure sign-in pop-up when needed).

Installation

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

Quick start

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

```
// src/Home.jsx
import React, { useMemo, useState } from 'react';
import { WalletPaymentForm } from 'evzone-pay-ss9';

const PUBLISHABLE_KEY = 'pk_test_123';
```

```

const ENTERPRISE_WALLET_NO = 'EVZ-123456';

export default function Home() {
  const [open, setOpen] = useState(false);

  const txn = useMemo(
    () => ({
      amount: 60000,
      type: 'Purchase',
      particulars: '2x Laptops, 1x Mouse',
      merchantLogo:
'https://res.cloudinary.com/dlfa42ans/image/upload/v1743854843/logo5_ljusns.png',
    }),
    []
  );

  return (
    <div style={{ padding: 24 }}>
      <h1>EVzone Checkout</h1>
      <p>Amount: UGX {txn.amount.toLocaleString()}</p>

      <button onClick={() => setOpen(true)}>Pay with EVzone Wallet</button>

      {open && (
        <WalletPaymentForm
          // REQUIRED (safe to expose)
          publishableKey={PUBLISHABLE_KEY}
          enterpriseWalletNo={ENTERPRISE_WALLET_NO}
          amount={txn.amount}
          type={txn.type}
          particulars={txn.particulars}
          merchantLogo={txn.merchantLogo}

          //contact shown in the "Cannot Continue" modal
          supportEmail="support@yourshop.com"
          supportPhone="+256 700 000000"

          onClose={() => setOpen(false)}
          onSuccess={(payload) => {
            console.log('Payment success:', payload);
            // Keep or close your wrapper/modal as you prefer.
          }}
        />
      )}
    </div>
  );
}

```

What happens at runtime? The SDK checks whether the shopper is already signed in to EVzone in this browser. If not, it prompts a secure pop-up sign-in and then continues the checkout automatically—your app

doesn't handle or see the user's number.

Troubleshooting (host app)

- **Form doesn't start** Ensure `publishableKey`, `enterpriseWalletNo`, `amount`, `type`, `particulars`, and `merchantLogo` are provided. If it still can't start, users will see a **friendly** "Cannot Continue" message with your `supportEmail`/`supportPhone` if you supplied them.

License & Support

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