

## BLOG 10 — Understanding PayCodes (EIP-681 for Safe Commerce)

**Category:** Developer Guides

**Length:** ~550 words

---

### **\*\*EIP-681 Made Crypto Payable.**

PayCodes Make Crypto Commercial.\*\*

EIP-681 (Ethereum Payment Request URIs) was a brilliant idea for its time.

It allowed wallets to embed:

- address
- token
- amount

into a simple payment link.

But commerce needs much more than this:

- undo windows
- confirmation logic
- fee stability
- metadata
- escrow flows
- cross-chain support
- workflow instructions
- transaction constraints

EIP-681 is too simple for real-world money movement.

So Dendrites created **PayCodes** — a modern, Safe Commerce–optimized evolution of payment links.

---

### **What Are PayCodes?**

PayCodes are enhanced payment request codes that allow a sender to execute Safe Commerce flows with a single tap.

They can be represented as:

- URL
- QR code
- short code
- deeplink
- URI

Every PayCode encodes:

- **amount** (ex: 25 USDC)
- **asset**
- **chain**
- **fee tier** (instant / eco)
- **Safe Commerce features**
  - UNDO
  - AckPay
  - Escrow
- **metadata**
- **merchant reference**
- **optional expiry**

This makes PayCodes not just a payment request — but a **transaction instruction set**.

---

## What PayCodes Solve

### Error-Proof Payments

Receivers can embed exactly what the sender must pay.

No more:

- wrong amount
- wrong asset
- wrong chain
- wrong address

Users stop making irreversible mistakes.

---

## **2 Safe Commerce Defaults**

PayCodes can include:

- undo\_window
- require\_ack
- escrow\_workflow

This means **every payment link can be safe by default.**

Example:

- QR code at a café
- Invoice link for a freelancer
- Delivery app payment prompt
- Marketplace checkout link

No custom contract needed.

---

## **3 Universally Compatible**

PayCodes work across:

- wallets
- apps
- browsers
- mobile

- QR scanners
- API calls

Any system that understands a URL can trigger a PayCode.

---

#### **Merchant-Grade Metadata**

PayCodes can embed metadata such as:

- order\_id
- merchant\_id
- shipping\_id
- invoice\_id
- subscription\_cycle

This enables clean reconciliation and integration with existing systems.

---

#### **Example PayCode (Readable Format)**

```
{  
  "to": "0xAbC123...",  
  "amount": "15 USDC",  
  "chain": "base",  
  "features": {  
    "undo": true,  
    "ack": true,  
    "escrow": false  
  },  
  "metadata": {  
    "orderId": "A1345",  
    "merchant": "NexaCoffee"
```

```
}  
}
```

---

### Generating a PayCode (Using the SDK)

```
const paycode = await dendrites.paycodes.generate({  
  to: receiverAddress,  
  amount: "15 USDC",  
  asset: "USDC",  
  chain: "base",  
  features: {  
    undo: true,  
    ack: false,  
    escrow: false  
  },  
  metadata: {  
    orderId: "A1345"  
  }  
});
```

Response:

```
{  
  "paycode": "dndx:base:pay:0xabc123?...",  
  "link": "https://pay.dendrites.ai/abc123",  
  "qr": "data:image/png;base64,..."  
}
```

You can display the QR, embed the link, or encode it in a button.

---

## How PayCodes Power Real Use Cases

### **Coffee shops & retail**

Tap > UNDO-protected payment > done.

### **E-commerce checkout**

Embed AckPay → merchant must confirm.

### **Freelancers & creators**

Milestone escrow encoded into a PayCode.

### **Deliveries & logistics**

Delivery completed → receiver accepts payment.

### **Cross-border remittances**

Users never touch chains, addresses, or tokens.

### **Subscription models**

PayCodes include metadata for recurring billing.

---

## PayCodes + Safe Commerce = Frictionless Payments

PayCodes are not just a link.

They are:

- safer
- structured
- predictable
- feature-rich
- developer-friendly

They bring all Safe Commerce features into one portable format that can travel anywhere a URL can travel.

This is how crypto becomes easy, human-friendly, and merchant-ready.