Documentación Técnica – LuminPay

# 1. Descripción General

El LuminPay es una aplicación web desarrollada con HTML, CSS y JavaScript puro que permite:

- Recibir datos del usuario (variable, nombre y monto).

- Generar múltiples URLs dinámicas ILP en diferentes monedas.

- Copiar dichas URLs al portapapeles con un clic.

- Simular (placeholder) la conexión con una wallet digital.

Este proyecto está diseñado como un prototipo educativo, cuyo objetivo es explorar el uso del Interledger Protocol (ILP) en la generación de rutas de pago.

# 2. Arquitectura del Sistema

El proyecto es completamente frontend y se estructura de la siguiente manera:

- HTML (index.html): define el formulario, la estructura de la interfaz y los contenedores dinámicos de resultados.

- CSS (incrustado en <style>): proporciona un diseño moderno con gradientes, sombras, animaciones y efectos de interacción.

- JavaScript (incrustado en <script>): contiene la lógica para procesar el formulario, generar URLs, manipular el DOM y copiar texto al portapapeles.

# 3. Interfaz de Usuario

**Componentes principales:**

- Formulario (<form id="ilpForm">)

- Campos: variableILP, userName, amount.

- Botón: Generar URLs ILP.

- Sección de resultados (<div id="result">)

- Nombre del usuario.

- Monto ingresado.

- Lista de monedas soportadas.

- Botones de copiado de cada URL.

- Botón: Identificar mi cartera.

**Diseño visual:**

- Fondo con gradiente (violeta → azul).

- Contenedor con efecto glassmorphism.

- Animaciones de entrada (slideIn).

- Botones con hover y sombras dinámicas.

# 4. Lógica de Funcionamiento (JavaScript)

Variables principales:

const currencies = ["USD", "EUR", "XRP", "BTC", "ETH", "JPY", "MXN", "CAD", "AUD", "GBP"];

const form = document.getElementById("ilpForm");

const resultDiv = document.getElementById("result");

const currencyList = document.getElementById("currencyList");

const walletButton = document.getElementById("walletButton");

Flujo de ejecución:

1. Captura del formulario (form.addEventListener("submit")).

- Previene la recarga de la página.

- Lee valores de variableILP, userName y amount.

- Inserta nombre y monto en resultados.

- Limpia lista previa de monedas.

2. Generación dinámica de URLs.

- Itera sobre currencies y construye cada URL con:

https://ilp.interledger-test.dev/{variableILP}?amount={amount}&currency={currency}

- Crea un <li> con la URL y un botón de copiado.

3. Manejo del botón "Identificar mi cartera".

- Muestra un alert con funcionalidad futura.

# 5. Funcionalidades Clave

- Generación de múltiples URLs ILP basadas en entradas del usuario.

- Manejo dinámico del DOM para mostrar resultados.

- Copiado directo de URLs al portapapeles.

- Diseño moderno, responsivo y atractivo.

- Futuro: integración real con una wallet digital para ejecutar pagos.

# 6. Tecnologías Usadas

Frontend:

- HTML5: estructura semántica.

- CSS3: gradientes, sombras, animaciones.

- JavaScript ES6: manejo del DOM, eventos y clipboard API.

Dependencias externas:

- Google Fonts (Roboto, Major Mono Display).

# 7. Seguridad

- navigator.clipboard requiere ejecución en contexto seguro (HTTPS o localhost).

- No se exponen claves privadas ni información sensible.

- Las URLs generadas son únicamente para pruebas educativas en entornos testnet.

# 8. Posibles Mejoras

- Implementar conexión real con Interledger y una wallet (ej. Rafiki Wallet).

- Añadir validaciones avanzadas (monto positivo, restricción de caracteres).

- Exportar URLs como archivo .txt o QR.

- Soporte de internacionalización (i18n).

- Migración a un framework moderno (React, Vue, Svelte) para escalabilidad.

# Technical Documentation – LuminPay

## 1. Overview

LuminPay is a web application developed with HTML, CSS, and pure JavaScript that allows you to:

- Receive user data (variable, name, and amount).

- Generate multiple dynamic ILP URLs in different currencies.

- Copy these URLs to the clipboard with one click.

- Simulate (placeholder) the connection with a digital wallet.

This project is designed as an educational prototype, whose objective is to explore the use of the Interledger Protocol (ILP) in the generation of payment routes.

## 2. System Architecture

The project is completely frontend and is structured as follows:

- HTML (index.html): defines the form, the interface structure, and the dynamic result containers.

- CSS (embedded in <style>): provides a modern design with gradients, shadows, animations, and interaction effects.

- JavaScript (embedded in <script>): contains the logic for processing the form, generating URLs, manipulating the DOM, and copying text to the clipboard.

## 3. User Interface

Main components:

- Form (<form id="ilpForm">)

- Fields: variableILP, userName, amount.

- Button: Generate ILP URLs.

- Results section (<div id="result">)

- User name.

- Amount entered.

- List of supported currencies.

- Copy buttons for each URL.

- Button: Identify my wallet.

Visual design:

- Gradient background (purple → blue).

- Container with glassmorphism effect.

- Entry animations (slideIn).

- Buttons with hover and dynamic shadows.

## 4. Operating Logic (JavaScript)

Main variables:

const currencies = [“USD”, “EUR”, “XRP”, “BTC”, “ETH”, “JPY”, “MXN”, “CAD”, ‘AUD’, “GBP”];

const form = document.getElementById(“ilpForm”);

const resultDiv = document.getElementById(“result”);

const currencyList = document.getElementById(“currencyList”);

const walletButton = document.getElementById(“walletButton”);

Execution flow:

1. Capture the form (form.addEventListener(“submit”)).

- Prevents the page from reloading.

- Reads values from variableILP, userName, and amount.

- Inserts name and amount into results.

- Clears previous currency list.

2. Dynamic URL generation.

- Iterates over currencies and builds each URL with:

https://ilp.interledger-test.dev/{variableILP}?amount={amount}&currency={currency}

- Creates an <li> with the URL and a copy button.

3. Handling of the “Identify my wallet” button.

- Display an alert with future functionality.

## 5. Key Features

- Generation of multiple ILP URLs based on user input.

- Dynamic DOM handling to display results.

- Direct copying of URLs to the clipboard.

- Modern, responsive, and attractive design.

- Future: real integration with a digital wallet to execute payments.

## 6. Technologies Used

Frontend:

- HTML5: semantic structure.

- CSS3: gradients, shadows, animations.

- JavaScript ES6: DOM handling, events, and clipboard API.

External dependencies:

- Google Fonts (Roboto, Major Mono Display).

## 7. Security

- navigator.clipboard requires execution in a secure context (HTTPS or localhost).

- Private keys and sensitive information are not exposed.

- The URLs generated are only for educational testing in testnet environments.

## 8. Possible Improvements

- Implement real connection with Interledger and a wallet (e.g., Rafiki Wallet).

- Add advanced validations (positive amount, character restriction).

- Export URLs as .txt or QR files.

- Internationalization support (i18n).

- Migration to a modern framework (React, Vue, Svelte) for scalability.