# Family Payment System

<p align="center">

<img src="https://raw.githubusercontent.com/interledger/open-payments/main/docs/public/img/logo.svg" width="700" alt="Open Payments">

</p>

## What's the problem?

Family Payment System is a simple web application that implements the Open Payments standard to facilitate instant money transfers between family members. The application serves different use cases including:

- Allowance payments from parents to children

- Chore payments and rewards

- P2P transfers within the family

- Educational tool for digital money concepts

- Quick family expense sharing

- Emergency money transfers

The Family Payment System implements the Open Payments APIs through a simple architecture:

- A \*\*Express.js backend\*\* which handles payment requests and integrates with Open Payments APIs

- A \*\*web interface\*\* that provides a user-friendly form for selecting recipients and amounts

- An \*\*Open Payments integration\*\* that creates incoming payments through the Interledger network using wallet addresses for family members

The application uses the Open Payments standard to ensure interoperability and leverages the Interledger Protocol for efficient, low-cost transfers between family wallets configured on the test network.

## What technology will they use?

The Family Payment System provides developers with a simple implementation using:

Currently, the application uses:

- [Node.js with Express.js](https://expressjs.com/) for the backend server

- [Open Payments Standard](https://openpayments.dev/) for payment processing

- [Interledger Protocol](https://interledger.org/) for the underlying payment network

- [Vanilla JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript) for the frontend interface

with plans to expand functionality and add more family-oriented features in the future.

## New to Open Payments?

Never heard of Open Payments before? Or would you like to learn more? Here are some excellent places to start:

- [Open Payments Website](https://openpayments.dev/)

- [Interledger Foundation](https://interledger.org/)

- [Open Payments Specifications](https://github.com/interledger/open-payments-specifications)

- [Interledger Explainer Video](https://twitter.com/Interledger/status/1567916000074678272)

- [Web monetization](https://webmonetization.org/)

###Project Architecture

##The Problem

Families need a simple and direct way to transfer money between members, especially for situations like allowances, payments for household chores, or transfers between parents and children. Traditional payment methods are often complicated, expensive, or require multiple steps.

##The Technology

The project uses the following technologies:

- \*\*Backend\*\*: Node.js with Express.js

- \*\*Payment API\*\*: Open Payments Standard (Interledger)

- \*\*Frontend\*\*: HTML/CSS/JavaScript (vanilla)

- \*\*Protocol\*\*: ILP (Interledger Protocol) for transactions

- \*\*Test Environment\*\*: Interledger Test Network

##The Solution

A simple web application that allows family members to make instant payments using the Open Payments standard. The application provides an intuitive interface where users can:

- Select the recipient (parent/child)

- Specify the amount to transfer

- Execute the payment with a single click

##The Benefits

- \*\*Simplicity\*\*: Family-friendly interface easy to use for all members

- \*\*Speed\*\*: Instant payments without complex intermediaries

- \*\*Transparency\*\*: Clear tracking of family transactions

- \*\*Financial Education\*\*: Tool for teaching digital money concepts to children

- \*\*Interoperability\*\*: Based on open standards that work with multiple providers

- \*\*Low Cost\*\*: Leverages Interledger Protocol efficiency

##Architecture Stack

```

Frontend (HTML/CSS/JS)

↓

Express.js Server

↓

Open Payments API

↓

Interledger Network

↓

Wallet Providers

```

##Essential Functions

\*\*Core Functions:\*\*

- \*\*Wallet Selection\*\*: Dropdown to choose recipient (parent/child1)

- \*\*Amount Input\*\*: Numeric field to specify amount in USD

- \*\*Payment Processing\*\*: Creation of incoming payments via Open Payments API

- \*\*Error Handling\*\*: Validation and error feedback

- \*\*Visual Response\*\*: Confirmation of successful payments

\*\*Security Functions:\*\*

- Amount validation

- Valid wallet verification

- Secure handling of authorization tokens

- Error logging for debugging

##Development Responsibilities

\*\*Backend Development:\*\*

- Backend API (`/pagar` endpoint)

- Open Payments Integration (create incoming payments)

- Wallet Configuration (URL mapping)

- Error Handling and logging

- Express Server and middleware (CORS, JSON parsing)

\*\*Frontend Development:\*\*

- User Interface (payment form)

- Client-side validation

- Backend API integration

- Visual feedback for users

\*\*DevOps Configuration:\*\*

- Environment Variables (ILP\_TOKEN)

- Deployment and server configuration

- Testing on Interledger Testnet

###Local Development Environment

##Prerequisites

- [Node.js](https://nodejs.org/) (version 14 or higher)

- npm or yarn package manager

- Access token for Interledger Test Network

##Environment Setup

```sh

# install dependencies

npm install express cors node-fetch

# create environment file

echo "ILP\_TOKEN=your\_token\_here" > .env

# install additional dependencies if needed

npm install dotenv

```

##Local Development

You can start the application by running the command

```sh

npm start

```

The application will be available at http://localhost:3000

##Project Structure

```

family-payment-system/

├── server.js # Main Express server

├── index.html # Application frontend

├── package.json # Project dependencies

└── .env # Environment variables (don't include in git)

```

##Usage Instructions

1. Open http://localhost:3000 in your browser
2. Select the recipient (parent or child1)
3. Enter the amount in USD
4. Click "Pay"
5. View payment confirmation

##Useful commands

```sh

# start the server:

npm start

# install dependencies:

npm install

# check application logs:

node server.js

```

###Contributing

Please read the contribution guidelines before submitting contributions. All contributions must adhere to our code of conduct.

##Open Payments Integration

This application uses the Open Payments standard to facilitate payment interoperability. For more information about Open Payments, visit [openpayments.dev](https://openpayments.dev).

**Sistema de Pagos Familiares**

<p align="center"> <img src="https://raw.githubusercontent.com/interledger/open-payments/main/docs/public/img/logo.svg" width="700" alt="Open Payments"> </p>

**¿Cuál es el problema?**

El Sistema de Pagos Familiares es una aplicación web simple que implementa el estándar Open Payments para facilitar transferencias de dinero instantáneas entre miembros de la familia. La aplicación sirve diferentes casos de uso incluyendo:

* Pagos de mesadas de padres a hijos
* Pagos de tareas domésticas y recompensas
* Transferencias P2P dentro de la familia
* Herramienta educativa para conceptos de dinero digital
* Compartir gastos familiares rápidamente
* Transferencias de dinero de emergencia

El Sistema de Pagos Familiares implementa las APIs de Open Payments a través de una arquitectura simple:

* Un **backend Express.js** que maneja las solicitudes de pago e integra con las APIs de Open Payments
* Una **interfaz web** que proporciona un formulario fácil de usar para seleccionar destinatarios y montos
* Una **integración Open Payments** que crea pagos entrantes a través de la red Interledger usando direcciones de billetera para los miembros de la familia

La aplicación utiliza el estándar Open Payments para asegurar interoperabilidad y aprovecha el Protocolo Interledger para transferencias eficientes y de bajo costo entre billeteras familiares configuradas en la red de prueba.

**¿Qué tecnología utilizarán?**

El Sistema de Pagos Familiares proporciona a los desarrolladores una implementación simple usando:

Actualmente, la aplicación utiliza:

* [Node.js con Express.js](https://expressjs.com/) para el servidor backend
* [Estándar Open Payments](https://openpayments.dev/) para el procesamiento de pagos
* [Protocolo Interledger](https://interledger.org/) para la red de pagos subyacente
* [JavaScript Vanilla](https://developer.mozilla.org/en-US/docs/Web/JavaScript) para la interfaz frontend

con planes de expandir la funcionalidad y agregar más características orientadas a la familia en el futuro.

**¿Nuevo en Open Payments?**

¿Nunca has escuchado sobre Open Payments antes? ¿O te gustaría aprender más? Aquí hay algunos excelentes lugares para comenzar:

* [Sitio Web de Open Payments](https://openpayments.dev/)
* [Fundación Interledger](https://interledger.org/)
* [Especificaciones de Open Payments](https://github.com/interledger/open-payments-specifications)
* [Video Explicativo de Interledger](https://twitter.com/Interledger/status/1567916000074678272)
* [Monetización Web](https://webmonetization.org/)

**Arquitectura del Proyecto**

**El Problema**

Las familias necesitan una forma simple y directa de transferir dinero entre miembros, especialmente para situaciones como mesadas, pagos por tareas domésticas, o transferencias entre padres e hijos. Los métodos de pago tradicionales son a menudo complicados, costosos, o requieren múltiples pasos.

**La Tecnología**

El proyecto utiliza las siguientes tecnologías:

* **Backend**: Node.js con Express.js
* **API de Pagos**: Estándar Open Payments (Interledger)
* **Frontend**: HTML/CSS/JavaScript (vanilla)
* **Protocolo**: ILP (Protocolo Interledger) para transacciones
* **Entorno de Prueba**: Red de Prueba Interledger

**La Solución**

Una aplicación web simple que permite a los miembros de la familia realizar pagos instantáneos usando el estándar Open Payments. La aplicación proporciona una interfaz intuitiva donde los usuarios pueden:

* Seleccionar el destinatario (padre/hijo)
* Especificar el monto a transferir
* Ejecutar el pago con un solo clic

**Los Beneficios**

* **Simplicidad**: Interfaz familiar fácil de usar para todos los miembros
* **Velocidad**: Pagos instantáneos sin intermediarios complejos
* **Transparencia**: Seguimiento claro de transacciones familiares
* **Educación Financiera**: Herramienta para enseñar conceptos de dinero digital a los niños
* **Interoperabilidad**: Basado en estándares abiertos que funcionan con múltiples proveedores
* **Bajo Costo**: Aprovecha la eficiencia del Protocolo Interledger

**Pila de Arquitectura**

Frontend (HTML/CSS/JS)

↓

Servidor Express.js

↓

API Open Payments

↓

Red Interledger

↓

Proveedores de Billetera

**Funciones Esenciales**

**Funciones Principales:**

* **Selección de Billetera**: Menú desplegable para elegir destinatario (padre/hijo1)
* **Entrada de Monto**: Campo numérico para especificar monto en USD
* **Procesamiento de Pagos**: Creación de pagos entrantes vía API Open Payments
* **Manejo de Errores**: Validación y retroalimentación de errores
* **Respuesta Visual**: Confirmación de pagos exitosos

**Funciones de Seguridad:**

* Validación de montos
* Verificación de billetera válida
* Manejo seguro de tokens de autorización
* Registro de errores para depuración

**Responsabilidades de Desarrollo**

**Desarrollo Backend:**

* API Backend (endpoint /pagar)
* Integración Open Payments (crear pagos entrantes)
* Configuración de Billeteras (mapeo de URL)
* Manejo de Errores y registro
* Servidor Express y middleware (CORS, análisis JSON)

**Desarrollo Frontend:**

* Interfaz de Usuario (formulario de pago)
* Validación del lado del cliente
* Integración de API backend
* Retroalimentación visual para usuarios

**Configuración DevOps:**

* Variables de Entorno (ILP\_TOKEN)
* Despliegue y configuración del servidor
* Pruebas en Testnet Interledger

**Entorno de Desarrollo Local**

**Prerrequisitos**

* [Node.js](https://nodejs.org/) (versión 14 o superior)
* Administrador de paquetes npm o yarn
* Token de acceso para la Red de Prueba Interledger

**Configuración del Entorno**

# instalar dependencias

npm install express cors node-fetch

# crear archivo de entorno

echo "ILP\_TOKEN=tu\_token\_aquí" > .env

# instalar dependencias adicionales si es necesario

npm install dotenv

**Desarrollo Local**

Puedes iniciar la aplicación ejecutando el comando

npm start

La aplicación estará disponible en http://localhost:3000

**Estructura del Proyecto**

family-payment-system/

├── server.js # Servidor Express principal

├── index.html # Frontend de la aplicación

├── package.json # Dependencias del proyecto

└── .env # Variables de entorno (no incluir en git)

**Instrucciones de Uso**

1. Abrir http://localhost:3000 en tu navegador
2. Seleccionar el destinatario (padre o hijo1)
3. Ingresar el monto en USD
4. Hacer clic en "Pagar"
5. Ver confirmación de pago

**Comandos útiles**

# iniciar el servidor:

npm start

# instalar dependencias:

npm install

# verificar registros de la aplicación:

node server.js

**Contribuciones**

Por favor lee las pautas de contribución antes de enviar contribuciones. Todas las contribuciones deben adherirse a nuestro código de conducta.

**Integración Open Payments**

Esta aplicación utiliza el estándar Open Payments para facilitar la interoperabilidad de pagos. Para más información sobre Open Payments, visita [openpayments.dev](https://openpayments.dev/).