

Affiliate Postback Tracking MVP

A full-stack affiliate tracking minimum viable product (MVP) for tracking conversions using Server-to-Server (S2S) postbacks. This system is reliable, privacy-friendly, and ideal for affiliate marketers and SaaS platforms seeking a robust tracking foundation.

Table of Contents

- [Overview](#)
 - [How Postback Tracking Works](#)
 - [Project Architecture](#)
 - [Database Schema](#)
 - [Installation & Getting Started](#)
 - [Using Docker \(Recommended\)](#)
 - [Local Development \(MySQL\)](#)
 - [API Endpoints](#)
 - [Frontend Usage](#)
 - [Troubleshooting](#)
 - [Development Structure](#)
 - [About](#)
-

Overview

This MVP enables affiliates and advertisers to reliably track clicks and conversions using S2S postbacks (no cookies or pixels). Includes:

- Click tracking and storage
 - Conversion postback endpoint
 - Affiliate dashboard for real-time reporting
 - Unique postback URL per affiliate
-

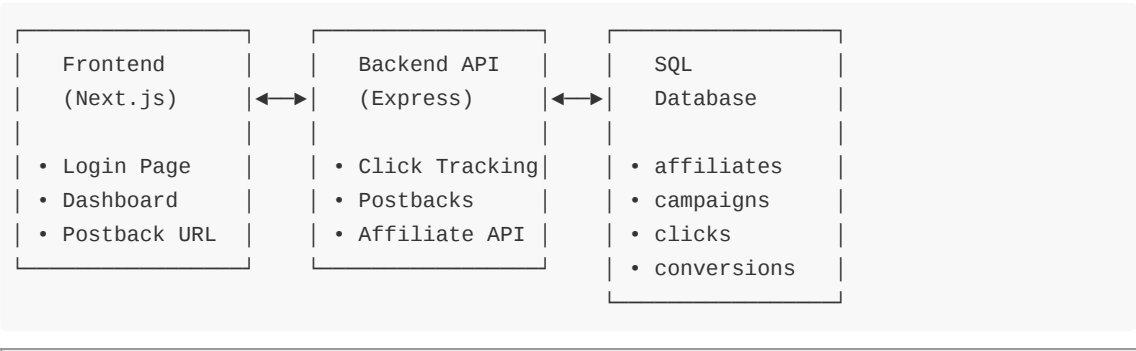
How Postback Tracking Works

- Click Tracking:**
User clicks affiliate's link. System records the click with a unique `click_id`.
- User Journey:**
User is redirected to the advertiser.
- Conversion Notification:**
When the user converts, advertiser server calls our `/postback` endpoint with the original `click_id` and details.
- Attribution:**
System finds the original click and attributes the conversion to the correct affiliate.

Advantages:

- Not affected by ad blockers, cookies, or browser limitations
 - Highly reliable and privacy-compliant
-

Project Architecture



Database Schema

Tables

affiliates

Column	Type	Notes
id	SERIAL PRIMARY KEY	
name	VARCHAR(100)	

campaigns

Column	Type	Notes
id	SERIAL PRIMARY KEY	
name	VARCHAR(100)	

clicks

Column	Type	Notes
id	SERIAL PRIMARY KEY	
affiliate_id	INT	REFERENCES affiliates(id)
campaign_id	INT	REFERENCES campaigns(id)
click_id	VARCHAR(100) UNIQUE	UNIQUE per affiliate+campaign
timestamp	TIMESTAMP	DEFAULT NOW()

conversions

Column	Type	Notes
id	SERIAL PRIMARY KEY	
click_id	INT	REFERENCES clicks(id)
amount	FLOAT	

currency	VARCHAR(10)	
timestamp	TIMESTAMP	DEFAULT NOW()

Installation & Getting Started

Prerequisites

- **Docker** and **Docker Compose** (recommended)
- **Node.js** v16+ and **npm**
- **MySQL** v8.0+ (for local/non-docker setup)

Using Docker (Recommended)

1. Clone the repository:

```
git clone https://github.com/your-username/affiliate-tracking-mvp.git
cd affiliate-tracking-mvp
```

2. Start all services:

```
docker-compose up --build
```

This launches:

- MySQL (port 3306)
- Backend API (Express, port 3001)
- Frontend (Next.js, port 3000)

Local Development (MySQL)

1. Clone the repository:

```
git clone https://github.com/your-username/affiliate-tracking-mvp.git
cd affiliate-tracking-mvp
```

2. Set up MySQL:

```
CREATE DATABASE affiliate_tracking;
CREATE USER 'affiliate_user'@'localhost' IDENTIFIED BY 'your_password';
GRANT ALL PRIVILEGES ON affiliate_tracking.* TO 'affiliate_user'@'localhost';
FLUSH PRIVILEGES;
```

3. Configure backend:

```
cd backend
cp .env.example .env
# Edit .env with DB_HOST, DB_USER, DB_PASSWORD, DB_NAME
```

4. Install dependencies:

```
cd backend
npm install
```

```
cd ../frontend
npm install
```

5. Run migrations:

```
cd ../backend
npm run migrate
# Creates tables and inserts sample data
```

6. Start servers:

- Backend: `cd backend && npm run dev`
- Frontend: `cd frontend && npm run dev`

Backend: <http://localhost:3001>

Frontend: <http://localhost:3000>

API Endpoints

Click Tracking

- **URL:** `/click`
- **Method:** `GET`
- **Query Params:** `affiliate_id`, `campaign_id`, `click_id`
- **Example:**
`http://localhost:3001/click?affiliate_id=1&campaign_id=10&click_id=abc123`
- **Response:**

```
{
  "success": true,
  "message": "Click tracked successfully",
  "data": {
    "click_id": "abc123",
    "affiliate_id": 1,
    "campaign_id": 10,
    "timestamp": "2023-08-23T15:30:45.123Z"
  }
}
```

Postback (Conversion)

- **URL:** `/postback`
- **Method:** `GET`
- **Query Params:** `affiliate_id`, `click_id`, `amount`, `currency`
- **Example:**
`http://localhost:3001/postback?affiliate_id=1&click_id=abc123&amount=100¤cy=USD`
- **Response:**

```
{
  "success": true,
  "message": "Conversion tracked successfully",
}
```

```
"data": {
  "conversion_id": 42,
  "click_id": "abc123",
  "affiliate_id": 1,
  "amount": 100,
  "currency": "USD",
  "timestamp": "2023-08-23T16:45:30.456Z"
}
```

Get Affiliate Clicks

- **URL:** /affiliates/:affiliate_id/clicks
- **Method:** GET
- **Example:**
http://localhost:3001/affiliates/1/clicks
- **Response:**

```
{
  "success": true,
  "data": [
    {
      "id": 1,
      "click_id": "abc123",
      "campaign_id": 10,
      "created_at": "2023-08-23T15:30:45.123Z"
    }
  ]
}
```

Get Affiliate Conversions

- **URL:** /affiliates/:affiliate_id/conversions
- **Method:** GET
- **Example:**
http://localhost:3001/affiliates/1/conversions
- **Response:**

```
{
  "success": true,
  "data": [
    {
      "id": 42,
      "click_id": "abc123",
      "amount": 100,
      "currency": "USD",
      "created_at": "2023-08-23T16:45:30.456Z"
    }
  ]
}
```

Frontend Usage

- Open <http://localhost:3000>
 - Select an affiliate from the dropdown
 - View:
 - Total clicks
 - Total conversions
 - Conversion rate
 - Recent click/conversion tables
 - Unique postback URL for the affiliate
-

Troubleshooting

- **Database connection errors:**
 - Check `.env` for correct credentials
 - Ensure database is running and accessible
 - **Port in use (EADDRINUSE):**
 - Stop the conflicting process or change port in `.env`
 - **Missing dependencies:**
 - Run `npm install` in `backend/` and `frontend/`
 - **Migration issues:**
 - Manually run SQL in `backend/migrations/schema.sql`
-

Development Structure

Backend (`backend/`)

- `server.js` - Express server
- `database.js` - MySQL connection
- `routes/`
 - `clicks.js` - Click tracking routes
 - `postbacks.js` - Conversion postback routes
 - `affiliates.js` - Affiliate data routes
- `migrations.sql` - Database schema
- `package.json`

Frontend (`frontend/`)

- `src/app/`
 - `page.tsx` - Login page
 - `dashboard/[affiliate_id]/page.tsx` - Dashboard
 - `postback-url/[affiliate_id]/page.tsx` - Postback URL display
 - `tailwind.config.js`
 - `package.json`
-

About

This is an open-source MVP for S2S affiliate conversion tracking.

- **Website:** affiliate-tracking-mvp.vercel.app
- **Contributions:** Welcome!
- **License:** MIT

Happy Tracking! 📊