# Event Processing Pipeline

This project simulates real-time event processing. It includes:  
- A Node.js API to receive events.  
- A Python API for enriching event data.

## Features:

- Event reception  
- Event enrichment with metadata (processed timestamp, location)

## Running the Project

### Node.js Server (Event Reception)

1. Navigate to the `server` folder.  
2. Install dependencies:  
 npm install  
3. Start the server:  
 node index.js

### Python Server (Event Enrichment)

1. Navigate to the `python-api` folder.  
2. (Optional) Set up a virtual environment:  
 python3 -m venv venv  
 source venv/bin/activate  
3. Install dependencies:  
 pip install flask  
4. Start the server:  
 python app.py

## Testing the Event Processing Pipeline

1. Send a POST request to the Node.js API to simulate an event:  
 curl -X POST http://localhost:3000/api/events \  
 -H "Content-Type: application/json" \  
 -d '{"eventType": "page\_view", "user": "test\_user", "timestamp": "2024-10-13T12:00:00Z"}'  
  
2. The Node.js API will pass the event to the Python API for enrichment and return the enriched event.

### Example output:

{  
 "original\_event": {  
 "eventType": "page\_view",  
 "user": "test\_user",  
 "timestamp": "2024-10-13T12:00:00Z"  
 },  
 "processed\_timestamp": "2024-10-13T13:00:00Z",  
 "location": "Charleston, SC"  
}

## Project Structure

- `server/`: Contains the Node.js server for event reception.  
- `python-api/`: Contains the Python server for event enrichment.