1. **🧾 1. initiate\_payment\_handler.py — 初始化支付請求**

import json

import os

import uuid

import datetime

import boto3

def lambda\_handler(event, context):

    body = json.loads(event['body'])

    order\_id = body['order\_id']

    user\_id = body['user\_id']

    amount = body['amount']

    payment\_method = body['payment\_method']

    # 產生交易 ID

    transaction\_id = str(uuid.uuid4())

    # 模擬呼叫第三方支付 API（實際應使用 requests.post）

    payment\_url = f"https://pay.example.com/checkout?tx={transaction\_id}"

    # 儲存交易紀錄（可改為寫入 RDS 或 DynamoDB）

    # 這裡僅示意

    print(f"Transaction created: {transaction\_id} for order {order\_id}")

    return {

        "statusCode": 200,

        "body": json.dumps({

            "transaction\_id": transaction\_id,

            "payment\_url": payment\_url,

            "expires\_at": (datetime.datetime.utcnow() + datetime.timedelta(minutes=15)).isoformat() + "Z"

        })

    }

1. **🔐 2. payment\_webhook\_handler.py — 處理支付結果並驗證 HMAC**

import json

import hmac

import hashlib

import os

# 假設密鑰儲存在環境變數或 AWS Secrets Manager

SECRET\_KEY = os.environ.get("PAYMENT\_SECRET\_KEY", "your-secret-key")

def verify\_signature(payload, signature):

    computed = hmac.new(

        SECRET\_KEY.encode('utf-8'),

        payload.encode('utf-8'),

        hashlib.sha256

    ).hexdigest()

    return hmac.compare\_digest(computed, signature)

def lambda\_handler(event, context):

    payload = event['body']

    headers = event['headers']

    signature = headers.get('X-Signature')

    if not verify\_signature(payload, signature):

        return {

            "statusCode": 403,

            "body": json.dumps({"error": "Invalid signature"})

        }

    data = json.loads(payload)

    transaction\_id = data['transaction\_id']

    status = data['status']

    gateway\_reference = data['gateway\_reference']

    # 更新交易狀態（可寫入 RDS 或 DynamoDB）

    print(f"Transaction {transaction\_id} updated to {status} with ref {gateway\_reference}")

    return {

        "statusCode": 200,

        "body": json.dumps({"message": "Webhook processed"})

    }

1. **🧩 整合建議**

* 使用 **API Gateway** 建立 /payments/initiate 與 /payments/webhook 路徑
* 使用 **Lambda + IAM Role** 控制存取權限
* 使用 **Secrets Manager** 儲存 PAYMENT\_SECRET\_KEY
* 使用 **CloudWatch Logs** 監控交易流程
* 可搭配 **DynamoDB 或 RDS** 儲存交易紀錄與訂單狀態